ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Απεικόνιση Σχεσιακού Μοντέλου σε Οντολογία Σημασιολογικού Ιστού

Πολυξένη Π. Κατσιούλη

Επιβλέπων: Ευστάθιος Π. Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΑΠΡΙΛΙΟΣ 2006

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Απεικόνιση Σχεσιακού Μοντέλου σε Οντολογία Σημασιολογικού Ιστού

Πολυξένη Π. Κατσιούλη

A.M.:1115200000028

ΕΠΙΒΛΕΠΩΝ:

Ευστάθιος Π. Χατζηευθυμιάδης, Επίκουρος Καθηγητής ΕΚΠΑ

ΠΕΡΙΛΗΨΗ

Η έλευση του Παγκόσμιου Ιστού έχει αλλάξει σημαντικά και ραγδαία τον τρόπο που οργανώνεται και διαμοιράζεται η πληροφορία. Η επόμενη γενιά του Παγκόσμιου Ιστού, ο Σημασιολογικός Ιστός, επιδιώκει να καταστήσει την πληροφορία πιο κατανοητή για τους υπολογιστές με την εισαγωγή μιας αυστηρότερης δομής βασισμένης στις οντολογίες. Με τον όρο οντολογία εννοούμε την ακριβή περιγραφή εννοιών καθώς και των σχέσεων που υπάρχουν ανάμεσά τους γύρω από ένα πεδίο ενδιαφέροντος. Ο Σημασιολογικός Ιστός, όμως, αντιμετωπίζει ένα πολύ σημαντικό πρόβλημα: την έλλειψη πραγματικών σημασιολογικών δεδομένων. Η παρούσα εργασία προτείνει μια μέθοδο δημιουργίας σημασιολογικών δεδομένων από δεδομένα αποθηκευμένα σε σχεσιακές βάσεις δεδομένων.

Είναι γνωστό ότι υπάρχει μεγάλη ποσότητα δεδομένων στον Ιστό αποθηκευμένη σε σχεσιακές βάσεις δεδομένων. Είναι λοιπόν πολύ σημαντικό να «παράγουμε» σημασιολογικά δεδομένα από σχεσιακά δεδομένα. Για να πραγματοποιηθεί ο εμπλουτισμός του Σημασιολογικού Ιστού με πραγματικά δεδομένα θα πρέπει πρώτα τα στοιχεία που αποτελούν το σχεσιακό σχήμα να αντιστοιχηθούν στα στοιχεία της οντολογίας. Στην παρούσα εργασία παρουσιάζεται μια μέθοδος εύρεσης πιθανών αντιστοιχίσεων, με ημι-αυτόματο τρόπο, ανάμεσα σε ένα σχεσιακό σχήμα και σε μια οντολογία που αφορούν στην ίδια θεματική περιοχή.

Το ταίριασμα δύο διαφορετικών σχημάτων είναι ένα αρκετά δύσκολο πρόβλημα αν αναλογιστεί κανείς τις διαφορές που υπάρχουν στους περιορισμούς που επιβάλλονται από το κάθε σχήμα καθώς και το γεγονός ότι τα δύο σχήματα μπορεί να έχουν σχεδιαστεί από διαφορετικά πρόσωπα και ως εκ τούτου είναι δυνατή η χρήση διαφορετικών όρων για την περιγραφή της ίδιας έννοιας. Η προτεινόμενη μεθοδολογία εκμεταλλεύεται αλγορίθμους εύρεσης της ομοιότητας ανάμεσα στα στοιχεία των δύο σχημάτων στοχεύοντας στην απλοποίηση της διαδικασίας.

Θεματική Περιοχή: εύρεση αντιστοιχίσεων μεταξύ στοιχείων δύο σχημάτων δεδομένων

Λέξεις Κλειδιά: βάσεις δεδομένων, σχεσιακό σχήμα, οντολογίες, ομοιότητα στοιχείων

ΠΕΡΙΕΧΟΜΕΝΑ

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

Σχήμα 6.13 Αποτελέσματα απλής αντιστοίχισης κλάσεων με τον αλγόριθμο της εικόνας 6.12 για $threshold_{Concept\ Mapping} = 0,5$ και $threshold_{Datatype\ property\ Mapping} = 0,5$.

Σχήμα 6.14 Αποτελέσματα αντιστοίχισης των πεδίων σε datatype-properties με τον αλγόριθμο της εικόνας 6.2 για $threshold_{Concept\ Mapping} = 0,5$ και $threshold_{Datatype\ property\ Mapping} = 0,5$.

Σχήμα 6.15 Αποτελέσματα απλής αντιστοίχισης κλάσεων με τον αλγόριθμο της εικόνας 6.2 για $threshold_{Concept\ Mapping} = 0,5$ και $threshold_{Datatype\ property\ Mapping} = 0,7$.

Σχήμα 6.16 Αποτελέσματα αντιστοίχισης των πεδίων σε datatype properties με τον αλγόριθμο της εικόνας 6.2 για $threshold_{Concept\ Mapping} = 0,5$ και $threshold_{Datatype\ property\ Mapping} = 0,7$.

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Η ανάπτυξη του Παγκοσμίου Ιστού (World Wide Web, WWW) έχει αλλάξει τον τρόπο επικοινωνίας των ανθρώπων και έχει προσφέρει στους χρήστες του ένα μεγάλο αριθμό από πηγές πληροφορίας. Ωστόσο, ενώ ο Παγκόσμιος Ιστός είναι κάτι πολύ συναρπαστικό για τους χρήστες, δεν συμβαίνει το ίδιο και με τους υπολογιστές καθώς αυτοί δεν μπορούν να κατανοήσουν την καταχωρημένη πληροφορία. Το νόημα της πληροφορίας είναι διαθέσιμο μόνο σε εκείνους που γνωρίζουν καλά τη γλώσσα στην οποία απεικονίζεται.

Τη λύση στα προβλήματα που αντιμετωπίζει ο Παγκόσμιος Ιστός έρχεται να δώσει μια επέκταση αυτού, εμπνευσμένη από τον Tim Berners Lee, ο λεγόμενος Σημασιολογικός Ιστός (Semantic Web, SW). Ο Σημασιολογικός Ιστός είναι το επόμενο βήμα του Παγκόσμιου Ιστού, όπου η πληροφορία αποκτά δομή και σημασιολογία ώστε να υποστηριχθεί η αποδοτική αναζήτηση, επεξεργασία και ενοποίηση δεδομένων. Σχεδιασμένος ως παγκόσμιο μέσο για την ανταλλαγή δεδομένων, βασίζεται στον ορισμό και την επαναχρησιμοποίηση κοινών λεξιλογίων από άτομα και κοινότητες, χαρακτηριστικά που τον καθιστούν προτιμητέο από άποψη κόστους για την καταγραφή και διάθεση γνώσης.

Ο Σημασιολογικός Ιστός βασίζεται στα μεταδεδομένα (metadata), ή μεταπληροφορία, τα οποία περιγράφουν τη σημασιολογία του περιεχομένου του Ιστού. Οραματίζεται τον εμπλουτισμό των δεδομένων του Ιστού με σημασιολογία έτσι ώστε να είναι κατανοητά από τους υπολογιστές επιτρέποντας έτσι την εξαγωγή υπονοούμενης (implicit) γνώσης. Το κύριο στοιχείο για την επίτευξη αυτού του στόχου είναι οι οντολογίες. Οι οντολογίες προσφέρουν μια εννοιολογική θεώρηση ενός πεδίου ενδιαφέροντος κάνοντας τη γνώση επαναχρησιμοποιήσιμη και διαμοιραζόμενη.

Οι οντολογίες καθεαυτές, όσο απαραίτητες κι αν είναι για την ανάπτυξη του Σημασιολογικού Ιστού, αποτελούν έναν τρόπο μοντελοποίησης εννοιών και σχέσεων που υπάρχουν ανάμεσά τους. Μόνο στην περίπτωση που «εφοδιαστούν» με πραγματικά δεδομένα μπορούν να χρησιμοποιηθούν για τη δημιουγία βάσεων γνώσης. Το κύριο πρόβλημα που αντιμετωπίζει ο Σημασιολογικός Ιστός είναι η έλλειψη σημασιολογικών δεδομένων. Καθίσταται λοιπόν αναγκαία η εύρεση ενός τρόπου δημιουργίας σημασιολογικών δεδομένων από ήδη υπάρχοντα δεδομένα. Είναι γνωστό ότι μεγάλη ποσότητα δεδομένων στον ιστό είναι αποθηκευμένα σε σχεσιακές βάσεις δεδομένων. Η πληροφορία αυτή είναι γνωστή ως Deep Web [49], σε αντίθεση με τον «επιφανειακό ιστό» (Surface Web) που αποτελείται από απλές στατικές ιστοσελίδες. Για να χρησιμοποιήσουμε τα δεδομένα των βάσεων δεδομένων στο Σημασιολογικό Ιστό είναι απαραίτητη η χρήση ενός μηχανισμού που θα αντιστοιχίζει τα στοιχεία του σχεσιακού σχήματος στα στοιχεία μιας οντολογίας και με βάση αυτές τις αντιστοιχίσεις θα εμπλουτίζει την οντολογία με δεδομένα από τη σχεσιακή βάση δεδομένων.

1.1 Στόχος της εργασίας

Στόχος της παρούσας εργασίας είναι η ανάπτυξη μιας μεθοδολογίας, με το όνομα RONTO, η οποία θα παράγει σημασιολογικά δεδομένα από σχεσιακά δεδομένα με ημι-αυτόματο τρόπο. Το πιο ενδιαφέρον και ουσιώδες βήμα αυτής της μεθοδολογίας είναι η απεικόνιση του σχεσιακού μοντέλου στην οντολογία, ή αλλιώς η εύρεση αντιστοιχίσεων ανάμεσα στα στοιχεία του σχεσιακού σχήματος και στα στοιχεία της οντολογίας. Η διαδικασία αυτή είναι αρκετά δύσκολη αν σκεφτεί κανείς ότι η βάση δεδομένων και η οντολογία, εν γένει, έχουν σχεδιαστεί από διαφορετικά άτομα, και παρά το ότι αναφέρονται στο ίδιο πεδίο, υπάρχουν διαφορές στην δομή, στην εκφραστικότητα, στα ονόματα των στοιχείων κ.ό.κ. Στη δύσκολη ανάπτυξη μιας τέτοιας μεθοδολογίας συμβάλλει και το γεγονός ότι τα δύο σχήματα (σχεσιακό και οντολογία) περιλαμβάνουν διαφορετικούς περιορισμούς λόγω της διαφορετικής εκφραστικότητας τους.

Η προτεινόμενη μεθοδολογία προτείνει αντιστοιχίσεις ανάμεσα σε ένα σχεσιακό σχήμα, σχεδιασμένο σε ένα τυπικό εμπορικό Σύστημα Διαχείρισης Βάσεων Δεδομένων, και σε μια οντολογία, υλοποιημένη σε γλώσσα OWL (Web Ontology Language), υπό την επίβλεψη του χρήστη. Δε διαχειρίζεται μόνο οντολογίες οι οποίες αποτελούν απλές ιεραρχίες εννοιών – κλάσεων – (taxonomies), αλλά και οντολογίες που περιέχουν αξιώματα και περιορισμούς (axiomatized ontologies). Ο χρήστης δεν είναι απαραίτητο

να έχει σχεδιάσει κάποιο από τα δύο μοντέλα ενώ η παρέμβασή του κατά τη διάρκεια απεικόνισης του σχεσιακού μοντέλου στην οντολογία είναι η ελάχιστη δυνατή.

1.2 Το σύστημα RONTO

Στην ενότητα αυτή παρουσιάζεται μια συνοπτική περιγραφή της λειτουργικότητας του RONTO.

Το RONTO είναι μια μεθοδολογία και ένα ημι-αυτόματο εργαλείο, στόχος του οποίου είναι:

- η απεικόνιση των στοιχείων ενός σχεσιακού μοντέλου στα στοιχεία μιας οντολογίας Σημασιολογικού Ιστού και
- η «μετακίνηση» των σχεσιακών δεδομένων, που είναι αποθηκευμένα στη βάση δεδομένων, σε στιγμιότυπα (instances) της οντολογίας.

Οι δύο παραπάνω στόχοι του συγκεκριμένου εργαλείου είναι γνωστοί ως *schema matching* και *data migration* αντίστοιχα.

Το RONTO προσανατολίζεται σε περιπτώσεις στις οποίες και η σχεσιακή βάση δεδομένων και η οντολογία προϋπάρχουν. Αυτό σημαίνει ότι τα στοιχεία του σχήματος της βάσης δεδομένων αντιστοιχίζονται στα στοιχεία της οντολογίας (π.χ. οι πίνακες της βάσης αντιστοιχίζονται σε κλάσεις της οντολογίας) και αν δεν υπάρχει στοιχείο στην οντολογία που να μπορεί να αντιστοιχηθεί με κάποιο στοιχείο της βάσης δεν δημιουργείται. Με άλλα λόγια τα δύο προς αντιστοίχιση σχήματα (σχεσιακό και οντολογία) δεν τροποποιούνται αλλά παραμένουν ως έχουν.

Η μεθοδολογία που ακολουθείται από το RONTO στοχεύει στη χρήση των περιορισμών και των ιδιοτήτων των προς αντιστοίχιση σχημάτων (σχεσιακό και εννοιολογικό) προκειμένου οι αντιστοιχίσεις που θα προτείνει να είναι όσο το δυνατόν πιο κοντά στις πραγματικές ενώ οι οντολογίες που υποστηρίζει είναι γραμμένες σε OWL και δεν αποτελούν απλές ταξινομίες αλλά ενδέχεται να περιέχουν αξιώματα και περιορισμούς.

1.3 Περιγραφή εργασίας

Η παρούσα εργασία εστιάζεται στην περιγραφή της μεθοδολογίας με την οποία επιτυγχάνεται ο πρώτος στόχος του RONTO, δηλαδή στην απεικόνιση του σχεσιακού μοντέλου σε οντολογία Σημασιολογικού Ιστού. Η οργάνωση της εργασίας έχει ως εξής.

Στο Κεφαλαίο 2 περιγράφεται η έννοια του σχεσιακού μοντέλου καθώς και τα βασικά χαρακτηριστικά του, ενώ στο κεφάλαιο 3 δίνονται στοιχεία σχετικά με τον Σημασιολογικό Ιστό και τα δομικά στοιχεία αυτού, δηλαδή τις οντολογίες. Περιγράφονται επίσης συνοπτικά οι δύο πιο γνωστές γλώσσες ανάπτυξης οντολογιών, η RDF(S) και η OWL.

Στο Κεφάλαιο 4 παρουσιάζονται αναλυτικά τα υπάρχοντα συστήματα και γλώσσες που έχουν αναπτυχθεί με στόχο την εύρεση αντιστοιχίσεων ανάμεσα σε ένα σχεσιακό σχήμα και μια οντολογία. Στο τέλος του κεφαλαίου αυτού αναφέρεται ποια πρέπει να είναι τα χαρακτηριστικά ενός τέτοιου συστήματος έτσι ώστε να είναι όσο το δυνατόν πληρέστερο και αποτελεσματικό.

Το πιο σημαντικό κεφάλαιο είναι το Κεφάλαιο 5 στο οποίο παρουσιάζεται η μεθοδολογία που ακολουθεί το RONTO προκειμένου να αντιστοιχίσει τα στοιχεία ενός σχεσιακού σχήματος σε εκείνα της οντολογίας. Η διαδικασία αυτή έχει χωριστεί σε φάσεις, σε κάθε μία από τις οποίες δίνονται και οι αντίστοιχοι αλγόριθμοι που χρησιμοποιούνται.

Στο κεφάλαιο 6 αξιολογούνται κάποιοι από τους αλγορίθμους εύρεσης αντιστοιχίσεων ανάμεσα στα στοιχεία των προς αντιστοίχιση σχημάτων και εξάγονται συμπεράσματα σχετικά με την απόδοση τους.

Η παρούσα εργασία ολοκληρώνεται με το Κεφάλαιο 7 στο οποίο δίνονται τα συμπεράσματα της όλης μελέτης και κάποια «ανοικτά» για μελέτη θέματα που αφορούν στο συγκεκριμένο εξεταζόμενο πεδίο έρευνας.

ΚΕΦΑΛΑΙΟ 2

ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Το σχεσιακό μοντέλο (relational model) [1] πρωτοπαρουσιάστηκε από τον Ted Codd της IBM Research το 1970 σαν ένα γενικό μοντέλο δεδομένων και στη συνέχεια εξελίχθηκε από τους Chris Date και Hugh Darwen. Προσέλκυσε άμεσα το ενδιαφέρον λόγω της απλότητας και της μαθηματικής θεμελίωσής του. Η λέξη «σχεσιακό» προέρχεται από την έννοια της λέξης «σχέση» όπως αυτή χρησιμοποιείται στα μαθηματικά. Το σχεσιακό μοντέλο χρησιμοποιεί την έννοια της μαθηματικής σχέσης σαν δομικό στοιχείο και η θεωρητική του βάση είναι η θεωρία συνόλων και ο κατηγορηματικός λογισμός πρώτης τάξης.

Άλλα μοντέλα δεδομένων είναι το ιεραρχικό (hierarchical model) και το δικτυωτό (network model) τα οποία προηγήθηκαν του σχεσιακού μοντέλου και χρησιμοποιούνται από ορισμένα συστήματα ακόμα και σήμερα. Οι ιεραρχικές και δικτυωτές βάσεις δεδομένων προϋπήρχαν των σχεσιακών αλλά η περιγραφή τους σαν μοντέλα έγινε μετά τον ορισμό του σχεσιακού μοντέλου.

Στο κεφάλαιο αυτό περιγράφονται οι βασικές έννοιες και τα χαρακτηριστικά του σχεσιακού μοντέλου.

2.1 Γενικές έννοιες του σχεσιακού μοντέλου

Η βάση δεδομένων στο σχεσιακό μοντέλο παριστάνεται ως μια συλλογή από σχέσεις κάθε μια από τις οποίες μπορούμε να πούμε ότι μοιάζει με πίνακα [1]. Όταν μια σχέση αντιμετωπίζεται ως ένας πίνακας (table) τιμών, κάθε γραμμή στον πίνακα παριστάνει μια συλλογή από τιμές δεδομένων που σχετίζονται. Οι τιμές αυτές μπορούν να ερμηνευτούν ως τα στοιχεία εκείνα που περιγράφουν μια οντότητα ή συσχέτιση του πραγματικού κόσμου. Για παράδειγμα, ο φοιτητής ενός πανεπιστημίου είναι μια

οντότητα που περιγράφεται από το ονοματεπώνυμό του, τον αριθμό μητρώου του, το τμήμα στο οποία φοιτά, το έτος κ.ά.

Βασικό δομικό στοιχείο στο σχεσιακό μοντέλο αποτελεί το **πεδίο ορισμού** (domain), ή ο τύπος δεδομένων (data type). Ένας τρόπος για να προσδιοριστεί το πεδίο ορισμού μιας στήλης του πίνακα, είναι να προσδιοριστεί ένας τύπος δεδομένων από τον οποίο επιλέγονται οι τιμές δεδομένων που σχηματίζουν το πεδίο (στήλη). Μια σχέση (πίνακας) είναι ένα σύνολο από πλειάδες (γραμμές) που δεν περιέχει διπλότυπα ενώ η σειρά εμφάνισης των πλειάδων είναι άνευ σημασίας.

Ένα **σχήμα σχέσης** R (relational schema), που συμβολίζεται με $R(A_1, A_2,\ldots, A_n)$, αποτελείται από ένα όνομα σχέσης R και μία λίστα από γνωρίσματα $A_1, A_2, ..., A_n$. Κάθε γνώρισμα (attribute) A_i είναι το όνομα ενός ρόλου που παίζει κάποιο πεδίο ορισμού D στο σχήμα της σχέσης R. Το D είναι το πεδίο ορισμού του A_i και συμβολίζεται με $dom(A_i)$ Ο **βαθμός** μιας σχέσης (degree of a relation) είναι το πλήθος n των γνωρισμάτων του σχήματός της R, ενώ ο πληθικός αριθμός μιας σχέσης είναι ο αριθμός των πλειάδων της.

Μια σχέση (relation) ή μια κατάσταση σχέσης (λέγεται και στιγμιότυπο σχέσης) r του σχήματος σχέσης $R(A_1, A_2, \ldots, A_n)$ – συμβολίζεται με r(R) - είναι ένα σύνολο από n-πλειάδες $r = \{t_1, t_2, \ldots, t_m\}$. Κάθε n-πλειάδα t είναι μια διατεταγμένη λίστα από n τιμές $t = <v_1, v_2, \ldots, v_n>$, όπου κάθε τιμή v_i είναι ένα στοιχείο του $dom(A_i)$ ή μια ειδική τιμή null.

Στο σχεσιακό μοντέλο οι τιμές κάθε γνωρίσματος είναι ατομικές, δεν μπορούν δηλαδή να διαιρεθούν σε επιμέρους συστατικά. Αυτό σημαίνει ότι δεν επιτρέπονται σύνθετα και πλειότιμα γνωρίσματα. Τα πλειότιμα γνωρίσματα πρέπει να αναπαρασταθούν με ξεχωριστές σχέσεις και τα σύνθετα γνωρίσματα παριστάνονται μόνο με τα συστατικά τους απλά γνωρίσματα. Παράδειγμα σύνθετου γνωρίσματος είναι η *Διεύθυνση* μιας οντότητας η οποία αποτελείται από τα απλά γνωρίσματα *Οδός*, *Αριθμός*, *Πόλη* και *Ταχυδρομικός Κώδικας*. Η απαγόρευση πλειότιμων και σύνθετων γνωρισμάτων στο σχεσιακό μοντέλο είναι γνωστή ως πρώτη κανονική μορφή (first normal form).

2.2 Περιορισμοί στο σχεσιακό μοντέλο

Σε ένα σχεσιακό σχήμα βάσης δεδομένων υπάρχουν κάποιοι περιορισμοί, οι οποίοι πρέπει να ικανοποιούνται από οποιαδήποτε κατάσταση των σχέσεων της βάσης δεδομένων. Στους περιορισμούς αυτούς περιλαμβάνονται οι:

- περιορισμοί πεδίου ορισμού,

- περιορισμοί κλειδιού,
- περιορισμοί ακεραιότητας οντοτήτων,
- περιορισμοί αναφορικής ακεραιότητας και
- περιορισμοί σημασιολογικής ακεραιότητας.

Αυτοί οι περιορισμοί ονομάζονται περιορισμοί κατάστασης επειδή ορίζουν τους περιορισμούς που πρέπει να ικανοποιεί μια έγκυρη κατάσταση της βάσης δεδομένων.

2.2.1 Περιορισμοί Πεδίου Ορισμού

Οι περιορισμοί πεδίου ορισμού καθορίζουν, όπως αναφέρθηκε και στην προηγούμενη παράγραφο, ότι η τιμή κάθε γνωρίσματος Α πρέπει να είναι μια ατομική τιμή από το πεδίο ορισμού αυτού του γνωρίσματος. Πεδίο ορισμού μιας σχέσης μπορεί να αποτελεί κάποιος από τους καθιερωμένους τύπους δεδομένων, όπως είναι οι αριθμητικοί, οι χαρακτήρες, οι συμβολοσειρές, η ημερομηνία, η ώρα, τα χρονικά σημεία και χρηματικά ποσά. Υπάρχουν και άλλα πιθανά πεδία ορισμού τα οποία καθορίζονται από τον κατασκευαστή της βάσης δεδομένων.

2.2.2 Περιορισμοί κλειδιού

Μια σχέση αποτελείται από ένα σύνολο πλειάδων και ως εκ τούτου δεν επιτρέπει την ύπαρξη διπλότυπων. Αυτό σημαίνει ότι δεν μπορεί δύο πλειάδες να έχουν τον ίδιο συνδυασμό τιμών για όλα τα γνωρίσματά τους. Συνήθως σε ένα σχήμα σχέσης υπάρχουν υποσύνολα γνωρισμάτων των οποίων οι τιμές αρκούν για να καθορίσουν μοναδικά μια οντότητα του συνόλου. Ένα τέτοιο σύνολο λέγεται **υπερ-κλειδί** (super key). Σε ένα σχήμα σχέσης υπάρχει πάντα ένα υπερ-κλειδί: το σύνολο των γνωρισμάτων της σχέσης.

Παρόλο που το υπερ-κλειδί μιας σχέσης χαρακτηρίζει μοναδικά μια οντότητα αυτής, είναι πιθανόν να περιέχει πλεονάζοντα γνωρίσματα. Για το λόγο αυτό εισάχθηκε η έννοια του υποψηφίου κλειδιού ενός συνόλου οντοτήτων. Ένα **υποψήφιο κλειδί** (candidate key) μιας σχέσης είναι ένα υπερ-κλειδί του οποίου οποιοδήποτε υποσύνολο γνωρισμάτων δεν είναι υπερκλειδί. Αναλυτικότερα, ένα υποψήφιο κλειδί είναι ένα υπερ-κλειδί από το οποίο δεν μπορούμε να παραλείψουμε οποιοδήποτε γνώρισμα χωρίς να παραβιαστεί ο περιορισμός της μοναδικότητας. Σε μια σχέση υπάρχει πάντα ένα υποψήφιο κλειδί τουλάχιστον.

Από τα υποψήφια κλειδιά μιας σχέσης επιλέγεται ένα το οποίο αποτελεί τον βασικό αντιπρόσωπο των οντοτήτων στη βάση και ονομάζεται **πρωτεύον κλειδί** (primary key). Σε μια σχέση υπάρχει πάντα ένα πρωτεύον κλειδί του οποίου οι τιμές προσδιορίζουν τις πλειάδες αυτής.

2.2.3 Περιορισμοί ακεραιότητας οντοτήτων

Ο περιορισμός ακεραιότητας οντοτήτων (entity integrity constraint) καθορίζει ότι δεν μπορεί η τιμή ενός πρωτεύοντος κλειδιού να είναι null. Αυτό ισχύει διότι η τιμή του πρωτεύοντος κλειδιού χρησιμοποιείται για να αναγνωριστεί η αντίστοιχη πλειάδα, έτσι αν μερικές πλειάδες έχουν τιμή null στο πρωτεύον κλειδί τους δεν μπορούν να αναγνωριστούν.

2.2.4 Περιορισμοί αναφορικής ακεραιότητας

Ένας περιορισμός αναφορικής ακεραιότητας (referential integrity constraint) ορίζεται μεταξύ δύο σχέσεων και χρησιμοποιείται για τη διατήρηση της συνέπειας μεταξύ των πλειάδων των δύο σχέσεων. Πιο απλά, ο περιορισμός αναφορικής ακεραιότητας ορίζει ότι μια πλειάδα μιας σχέσης που αναφέρεται σε μια άλλη σχέση πρέπει να αναφέρεται σε μια υπαρκτή πλειάδα αυτής της άλλης σχέσης.

Για έναν πιο αυστηρό ορισμό της αναφορικής ακεραιότητας εισάγουμε την έννοια του **ξένου κλειδιού**. Ένα σύνολο γνωρισμάτων FK στο σχήμα σχέσης R_1 είναι ξένο κλειδί της R_1 αν:

- Τα γνωρίσματα στο FK έχουν το ίδιο πεδίο ορισμού με τα γνωρίσματα του πρωτεύοντος κλειδιού ενός άλλου σχήματος σχέσης R_2. Τα γνωρίσματα στο FK λέμε ότι αναφέρονται στη σχέση R_2.
- Η τιμή του FK σε μια πλειάδα της R_1 είτε εμφανίζεται ως τιμή του πρωτεύοντος κλειδιού σε κάποια πλειάδα της R_2 είτε είναι null.

Συνήθως, οι περιορισμοί αναφορικής ακεραιότητας προκύπτουν από συσχετίσεις μεταξύ των οντοτήτων που παριστάνονται από τα σχήματα σχέσεων. Ένα ξένο κλειδί μπορεί να αναφέρεται στην ίδια του τη σχέση.

2.2.5 Περιορισμοί σημασιολογικής ακεραιότητας

Οι περιορισμοί σημασιολογικής ακεραιότητας (semantic integrity constraints) χρειάζεται να οριστούν και να επιβληθούν σε μια σχεσιακή βάση δεδομένων με τη χρήση μιας γλώσσας προσδιορισμού περιορισμών γενικού σκοπού. Παραδείγματα τέτοιων περιορισμών είναι «ο μισθός ενός εργαζόμενου δεν μπορεί να υπερβαίνει το μισθό του προϊσταμένου του» και «ο αριθμός των μαθημάτων που μπορεί να δηλώσει ένας φοιτητής του $1^{ου}$ εξαμήνου δεν μπορεί να είναι μεγαλύτερος του 6».

2.3 Σχεσιακές Βάσεις Δεδομένων

Μια σχεσιακή βάση δεδομένων περιέχει πολλές σχέσεις, οι πλειάδες των οποίων συνδέονται κατά διαφορετικούς τρόπους. Ένα σχεσιακό σχήμα βάσης δεδομένων (relational database schema) S είναι ένα σύνολο από σχεσιακά σχήματα $S = \{R_1, R_2, \ldots, R_m\}$ και ένα σύνολο από περιορισμούς ακεραιότητας (integrity constraints) IC. Μια κατάσταση σχεσιακής βάσης δεδομένων (relational database state) DB του S είναι ένα σύνολο από καταστάσεις σχέσεων $DB = \{r_1, r_2, \ldots, r_m\}$ τέτοιο ώστε κάθε r_i να είναι ένα στιγμιότυπο της σχέσης R_i και οι καταστάσεις r_i να ικανοποιούν τους περιορισμούς ακεραιότητας που προσδιορίζονται στο IC.

2.4 Πράξεις της Σχεσιακής Άλγεβρας

Εκτός από τον ορισμό του σχεσιακού μοντέλου και των περιορισμών, ένα μοντέλο δεδομένων περιλαμβάνει και ένα σύνολο από πράξεις για τη διαχείριση των δεδομένων. Ένα βασικό σύνολο πράξεων του σχεσιακού μοντέλου αποτελούν τη σχεσιακή άλγεβρα.

Ο πίνακας 2.1 περιγράφει το σκοπό και τον τρόπο συμβολισμού κάθε μιας από τις πράξεις της σχεσιακής άλγεβρας [1].

Πίνακας 2.1 Πράξεις της Σχεσιακής Άλγεβρας

ΠΡΑΞΗ	**ΣΚΟΠΟΣ**	**ΣΥΜΒΟΛΙΣΜΟΣ**
Επιλογή	Επιλέγει όλες τις πλειάδες, από μια σχέση R, που ικανοποιούν τη συνθήκη επιλογής.	$\sigma_{<\text{συνθήκη επιλογής}>}(R)$
Προβολή	Παράγει μια νέα σχέση με μερικά μόνο γνωρίσματα της R και απομακρύνει τις διπλές πλειάδες.	$\pi_{<\text{λίστα γνωρισμάτων}>}(R)$

Θήτα Συνένωση	Παράγει όλους τους συνδυασμούς πλειάδων από τις R_1 και R_2 που ικανοποιούν τη συνθήκη συνένωσης.	$R_1 \bowtie_{<\text{συνθήκη συνένωσης}>} R_2$
Συνένωση Ισότητας	Παράγει όλους τους συνδυασμούς πλειάδων από τις R_1 και R_2 που ικανοποιούν μια συνθήκη συνένωσης με συγκρίσεις ισότητας μόνο.	$R_1 \bowtie_{<\text{συνθήκη συνένωσης}>} R_2$ ή $R_1 \bowtie_{<\text{γνωρίσματα συνένωσης 1}>,<\text{γνωρίσματα συνένωσης 2}>} R_2$
Φυσική Συνένωση	Ίδια με τη Συνένωση Ισότητας εκτός από το ότι τα γνωρίσματα συνένωσης της R_2 δεν περιλαμβάνονται στο αποτέλεσμα. Αν τα γνωρίσματα συνένωσης έχουν τα ίδια ονόματα δεν χρειάζεται να προσδιοριστούν.	$R_1 *_{<\text{συνθήκη συνένωσης}>} R_2$ ή $R_1 *_{<\text{γνωρίσματα συνένωσης 1}>,<\text{γνωρίσματα συνένωσης 2}>} R_2$, ή $R_1 * R_2$
Ένωση	Παράγει μια σχέση που περιέχει όλες τις πλειάδες που βρίσκονται στην R_1, ή στην R_2, ή και στην R_1 και στην R_2 . Οι R_1 και R_2 πρέπει να είναι συμβατές ως προς την ένωση.	$R_1 \cup R_2$
Τομή	Παράγει μια σχέση που περιέχει τις κοινές πλειάδες των R_1 και R_2. Οι R_1 και R_2 πρέπει να είναι συμβατές ως προς την ένωση.	$R_1 \cap R_2$
Διαφορά	Παράγει μια σχέση που περιέχει τις πλειάδες της R_1 που δεν βρίσκονται στην R_2. Οι R_1 και R_2 πρέπει να είναι συμβατές ως προς την ένωση.	$R_1 - R_2$
Καρτεσιανό Γινόμενο	Παράγει μια σχέση που περιέχει τα γνωρίσματα των R_1 και R_2 και περιλαμβάνει ως πλειάδες όλους τους δυνατούς συνδυασμούς πλειάδων των R_1 και R_2.	$R_1 \times R_2$

Διαίρεση	Παράγει μια σχέση R(X) που περιέχει όλες τις πλειάδες t[X] στην $R_1(Z)$ που εμφανίζονται στην R_1 σε συνδυασμό με κάθε πλειάδα από την $R_2(Y)$, όπου $Z=X \cup Y$	$R_1(Z) \div R_2(Y)$

2.5 Συναρτησιακές Εξαρτήσεις

Η συναρτησιακή εξάρτηση (functional dependency) είναι ένας περιορισμός μεταξύ δύο συνόλων γνωρισμάτων της βάσης δεδομένων. Μια συναρτησιακή εξάρτηση (ΣΕ) σε μια σχέση R, που συμβολίζεται με $R:X \rightarrow Y$, ανάμεσα σε δύο σύνολα γνωρισμάτων Z και Y οορίζει έναν περιορισμό στις πιθανές πλειάδες που μπορούν να συγκροτήσουν ένα στιγμιότυπο σχέσης r της R. Ο περιορισμός ορίζει ότι για κάθε δύο πλειάδες t_1 και t_2 του r, τέτοιες ώστε $t_1[X] = t_2[X]$, πρέπει να έχουμε και $t_1[Y] = t_2[Y]$. Αυτό σημαίνει ότι οι τιμές της συνιστώσας X μιας πλειάδας καθορίζουν μοναδικά (ή συναρτησιακά) τις τιμές της συνιστώσας Y. Λέμε επίσης ότι υπάρχει μια συναρτησιακή εξάρτηση από το X στο Y.

Μια συναρτησιακή εξάρτηση είναι ιδιότητα του σχήματος σχέσης R και όχι κάποιας συγκεκριμένης επιτρεπτής κατάστασης r του R. Γι'αυτό το λόγο δεν μπορεί να εξαχθεί αυτόματα από ένα στιγμιότυπο μιας σχέσης, αλλά πρέπει να οριστεί ρητά.

2.6 Εξαρτήσεις Εγκλεισμού

Οι εξαρτήσεις εγκλεισμού (inclusion dependencies) τυποποιούν περιορισμούς μεταξύ σχέσεων και όχι μεταξύ γνωρισμάτων της ίδιας σχέσης όπως οι συναρτησιακές εξαρτήσεις. Μια εξάρτηση εγκλεισμού (ΕΕ), συμβολιζόμενη ως R.X « S.Y, μεταξύ δύο συνόλων γνωρισμάτων – του X από ένα σχήμα σχέσης R και του Y από ένα σχήμα σχέσης S – προσδιορίζει τον περιορισμό ότι, κάθε φορά που το r είναι ένα στιγμιότυπο της σχέσης R και το s στιγμιότυπο της σχέσης S, θα πρέπει να έχουμε:

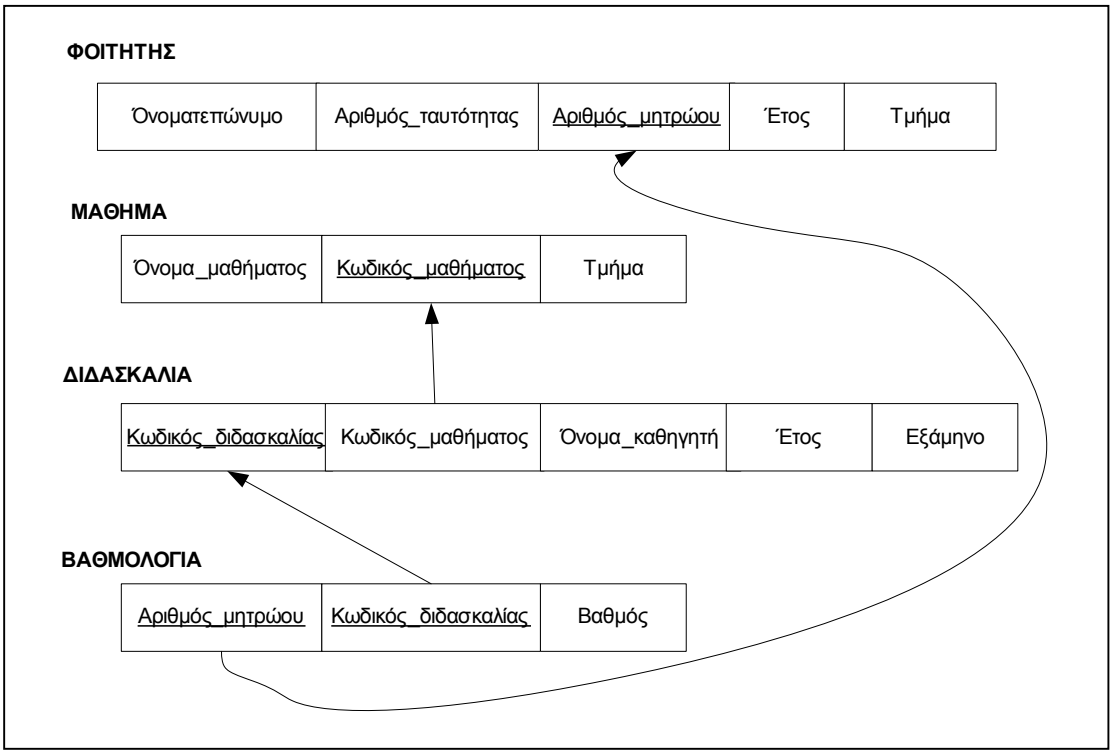
$\pi_X(r(R)) \subseteq \pi_Y(s(S))$

Προφανώς, τα σύνολα των γνωρισμάτων επί των οποίων προσδιορίζεται η εξάρτηση εγκλεισμού πρέπει να έχουν το ίδιο πλήθος γνωρισμάτων. Επιπλέον, τα πεδία ορισμού των αντίστοιχων γνωρισμάτων πρέπει να είναι συμβατά. Οι εξαρτήσεις εγκλεισμού μπορούν να χρησιμοποιηθούν για την αναπαράσταση περιορισμών αναφορικής ακεραιότητας αλλά και για να παραστήσουν συσχετίσεις κλάσης/υποκλάσης (class/subclass relationship).

2.7 Εφαρμογή

Στην ενότητα αυτή κάνουμε κάποιους από τους παραπάνω ορισμούς πιο κατανοητούς χρησιμοποιώντας ένα παράδειγμα βάσης δεδομένων που ονομάζεται ΠΑΝΕΠΙΣΤΗΜΙΟ και αποθηκεύει πληροφορίες για τους φοιτητές, τα μαθήματα και τη βαθμολογία τους σ'αυτά.

Στην εικόνα 2.1 δίνεται το διάγραμμα του σχεσιακού σχήματος της βάσης δεδομένων ΠΑΝΕΠΙΣΤΗΜΙΟ.



Εικόνα 2.1 Διάγραμμα σχεσιακού σχήματος για τη βάση δεδομένων ΠΑΝΕΠΙΣΤΗΜΙΟ

Το παραπάνω σχεσιακό σχήμα βάσης δεδομένων περιέχει 4 σχέσεις (πίνακες). Τα σχήματα των σχέσεων αυτών είναι τα ακόλουθα:

- ΦΟΙΤΗΤΗΣ (Ονοματεπώνυμο, Αριθμός_μητρώου, Έτος, Τμήμα)
- ΜΑΘΗΜΑ (Όνομα_μαθήματος, Κωδικός_μαθήματος, Τμήμα)
- ΔΙΔΑΣΚΑΛΙΑ (Κωδικός_διδασκαλίας, Κωδικός_μαθήματος, Όνομα_καθηγητή, Έτος, Εξάμηνο)
- ΒΑΘΜΟΛΟΓΙΑ (Αριθμός_μητρώου, Κωδικός_διδασκαλίας, Βαθμός)

Τα υπογραμμισμένα γνωρίσματα σε κάθε σχέση αποτελούν το πρωτεύον κλειδί της. Έτσι, πρωτεύον κλειδί για τον πίνακα των φοιτητών αποτελεί ο αριθμός μητρώου, για τα μαθήματα και τις διδασκαλίες ένας κωδικός που τις καθορίζει μοναδικά ενώ ο πίνακας με τις βαθμολογίες έχει πρωτεύον κλειδί τα γνωρίσματα Αριθμός_μητρώου και Κωδικός_διδασκαλίας.

Οι περιορισμοί αναφορικής ακεραιότητας παρουσιάζονται με τη χρήση των κατευθυνόμενων τόξων από το ξένο κλειδί στο πρωτεύον κλειδί της σχέσης στην οποία αναφέρονται. Στο παραπάνω σχεσιακό σχήμα υπάρχουν 3 ξένα κλειδιά:

- ο Αριθμός_μητρώου του πίνακα ΒΑΘΜΟΛΟΓΙΑ ο οποίος αναφέρεται στη στήλη Αριθμός_μητρώου του πίνακα ΦΟΙΤΗΤΗΣ,
- ο Κωδικός_διδασκαλίας του πίνακα ΒΑΘΜΟΛΟΓΙΑ που παίρνει τιμές από την ομώνυμη στήλη του πίνακα ΔΙΔΑΣΚΑΛΙΑ και
- ο Κωδικός_μαθήματος του πίνακα ΔΙΔΑΣΚΑΛΙΑ που αναφέρεται στο πρωτεύον κλειδί της σχέσης ΜΑΘΗΜΑ.

Ένα υποψήφιο κλειδί του πίνακα ΦΟΙΤΗΤΗΣ είναι ο Αριθμός_ταυτότητας καθώς η τιμή του είναι μοναδική για κάθε οντότητα του συνόλου των φοιτητών.

Στη συνέχεια δίνονται μερικά παραδείγματα πεδίων ορισμού:

- *Ονοματεπώνυμο:* Το σύνολο των ονοματεπώνυμων ανθρώπων.
- *Όνομα_καθηγητή:* Το σύνολο των ονοματεπώνυμων ανθρώπων.
- *Βαθμός:* Το σύνολο των φυσικών αριθμών μεταξύ 5 και 10, δηλαδή το {5, 6, 7, 8, 9, 10}.
- *Αριθμός_μητρώου:* Το σύνολο των επιτρεπόμενων 13ψήφιων αριθμών μητρώου.
- *Έτος:* Το σύνολο των φυσικών αριθμών που είναι μεγαλύτεροι από το 0.
- *Τμήμα:* Το σύνολο των ονομάτων των ακαδημαϊκών τμημάτων ενός πανεπιστημίου, όπως Φυσικό, Μαθηματικό, Χημικό κ.ά.
- *Εξάμηνο:* Το σύνολο των φυσικών αριθμών που είναι μεγαλύτεροι από το 0.

ΚΕΦΑΛΑΙΟ 3

ΣΗΜΑΣΙΟΛΟΓΙΚΟΣ ΙΣΤΟΣ

ΚΑΙ

ΟΝΤΟΛΟΓΙΕΣ

Η ανάπτυξη του Παγκοσμίου Ιστού [39] άλλαξε ριζικά τον τρόπο επικοινωνίας των ανθρώπων, τον τρόπο διάθεσης και ανάπτυξης της πληροφορίας καθώς και τον τρόπο διεξαγωγής των επιχειρηματικών δραστηριοτήτων. Η ανάπτυξη του Παγκοσμίου Ιστού δεν θα ήταν τόσο ραγδαία αν δεν υπήρχαν οι μηχανές αναζήτησης (π.χ. Google [40], Yahoo! [41], AltaVista [42]) που βασίζονται σε λέξεις-κλειδιά. Παρόλα αυτά υπάρχουν κάποια προβλήματα που σχετίζονται με τη χρήση τους. Ένα από αυτά είναι το γεγονός ότι τα αποτελέσματα που επιστρέφουν είναι απλές ιστοσελίδες. Έτσι αν χρειαζόμαστε πληροφορίες οι οποίες είναι κατανεμημένες σε ξεχωριστά έγγραφα θα πρέπει να εκτελέσουμε αρκετές αναζητήσεις με διαφορετικές λέξεις-κλειδιά προκειμένου να ανακτήσουμε τις σχετικές πληροφορίες. Ένα επίσης σημαντικό πρόβλημα των μηχανών αναζήτησης είναι η μεγάλη «ευαισθησία» τους στο λεξιλόγιο. Πολύ συχνά δεν επιστρέφονται έγγραφα που σχετίζονται με τις λέξεις-κλειδιά γιατί χρησιμοποιούν διαφορετική ορολογία. Αυτό δεν είναι αποτελεσματικό γιατί οι σημασιολογικά όμοιες αναζητήσεις θα έπρεπε να επιστρέφουν και ίδια αποτελέσματα. Τη λύση στα προβλήματα που αντιμετωπίζει ο Παγκόσμιος Ιστός έρχεται να δώσει μια επέκταση αυτού, ο Σημασιολογικός Ιστός.

Το κεφάλαιο αυτό χωρίζεται σε δύο ενότητες. Η πρώτη αφορά στον Σημασιολογικό Ιστό ενώ στη δεύτερη περιγράφονται οι οντολογίες οι οποίες αποτελούν το θεμελιώδες στοιχείο μοντελοποίησης του Σημασιολογικού Ιστού.

3.1 Σημασιολογικός Ιστός

Ο Σημασιολογικός Ιστός [31] (Semantic Web, SW) είναι ένα όραμα και μια πρόταση για την μετεξέλιξη του διαδικτύου και ειδικότερα του Παγκόσμιου Ιστού. Ο στόχος του Σημασιολογικού Ιστού είναι να εξελίξει το σημερινό διαδίκτυο έτσι ώστε οι πληροφορίες που υπάρχουν και διακινούνται σε αυτό να είναι κατανοητές, και κατ' επέκταση αυτόματα επεξεργάσιμες από τους υπολογιστές.

Ο Σημασιολογικός Ιστός δεν είναι ένας νέος Παγκόσμιος Ιστός. Είναι μια επέκταση και βελτίωση του σημερινού ιστού στην κατεύθυνση, κυρίως, της δόμησης της πληροφορίας έτσι ώστε να είναι προσπελάσιμη από προγράμματα υπολογιστών. Η σημερινή αναπαράσταση των κειμένων στις σελίδες του Ιστού που προορίζεται για χρήση από ανθρώπους θα αντικατασταθεί από αναπαράσταση κατανοητή στους υπολογιστές.

Ο Tim Berners-Lee, που επινόησε τον Παγκόσμιο Ιστό το 1989, είχε το όραμα ενός ιστού δεδομένων που μπορούν να επεξεργαστούν από μηχανές και έδωσε τον ακόλουθο ορισμό για τον Σημασιολογικό Ιστό.

The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation." [36]

Ο Σημασιολογικός Ιστός, που αποτελεί μια πρωτοβουλία της Κοινοπραξίας του Παγκοσμίου Ιστού (World Wide Web Consortium – W3C) [19], παρέχει μια διεθνώς προσβάσιμη πλατφόρμα που επιτρέπει σε αυτοματοποιημένα εργαλεία αλλά και σε ανθρώπους να επεξεργάζονται και να μοιράζονται δεδομένα.

Το κλειδί για την επίτευξη του παραπάνω στόχου είναι τα μεταδεδομένα (metadata) ή, αλλιώς, η μεταπληροφορία. Τα μεταδεδομένα κάνουν σαφή την πληροφορία που είναι αόριστη και την εκθέτουν προς αναζήτηση, επεξεργασία και ενοποίηση (integration). Τα μεταδεδομένα είναι δεδομένα που αναφέρονται σε άλλα δεδομένα (data about data). Συγκεκριμένα περιέχουν μέρος της σημασίας των δεδομένων, γεγονός που δικαιολογεί τον όρο «*σημασιολογικός»* στον Σημασιολογικό Ιστό.

3.1.1 Η Σχέση του Σημασιολογικού Ιστού με τον Παγκόσμιο Ιστό

Ο Παγκόσμιος Ιστός βασίζεται κυρίως σε έγγραφα γραμμένα σε HTML (Hypertext Markup Language) [5], μια γλώσσα η οποία περιγράφει το σώμα ενός δομημένου κειμένου δίνοντας έμφαση στην οπτική παρουσίαση, διανθίζοντάς το με αντικείμενα πολυμέσων όπως εικόνες και φόρμες διαλόγου.

Για παράδειγμα με τη χρήση της HTML και ενός προγράμματος πλοήγησης μπορούμε να δημιουργήσουμε και να παρουσιάσουμε μια ιστοσελίδα που απαριθμεί στοιχεία κάποιων προς πώληση βιβλίων. Όμως με την HTML δεν μπορεί να γίνει αντιληπτό ότι το στοιχείο “The Da Vinci Code” χαρακτηρίζει ένα βιβλίο αφού αναφέρεται στον τίτλο του ή ότι το στοιχείο “€20” αναφέρεται στην τιμή του. Δεν υπάρχει επίσης κανένας τρόπος να εκφραστεί το γεγονός ότι αυτά τα κομμάτια πληροφορίας είναι αλληλένδετα στην περιγραφή ενός συγκεκριμένου στοιχείου (δηλαδή ενός βιβλίου και μόνο), ευδιάκριτου από άλλα που ίσως απαριθμούνται στη σελίδα.

Ο Σημασιολογικός Ιστός αντιμετωπίζει την αδυναμία αυτή χρησιμοποιώντας γλώσσες που περιγράφουν δεδομένα και τη σχέση που έχουν αυτά μεταξύ τους. Δύο από αυτές τις γλώσσες είναι η RDF (Resource Description Framework) και OWL (Web Ontology Language) οι οποίες περιγράφονται στην ενότητα 3.2.3 και 3.2.4 αντίστοιχα και είναι κατανοητές από τους υπολογιστές.

3.1.2 Τα συστατικά του Σημασιολογικού Ιστού

Ο Σημασιολογικός Ιστός στηρίζεται από τις ακόλουθες γλώσσες και πρότυπα:

- *XML (Extensible Markup Language)* [9]*:* Είναι μια γλώσσα περιγραφής δεδομένων τα οποία είναι εύκολο να διαβαστούν και να επεξεργαστούν από ανθρώπους και προγράμματα. Δεν επιβάλλει κανέναν σημασιολογικό περιορισμό στα δεδομένα που περιγράφει.
- *XML Schema* [30]: Είναι μια γλώσσα η οποία περιορίζει τη δομή των XML εγγράφων.
- *RDF:* Είναι ένα μοντέλο περιγραφής και επεξεργασίας μεταδεδομένων.
- *RDF Schema:* Είναι ένας μηχανισμός περιγραφής πόρων και των σχέσεων ανάμεσα τους και αποτελεί σημασιολογική επέκταση του RDF.
- *OWL:* Παρέχει έναν τρόπο περιγραφής όρων και σχέσεων γύρω από ένα πεδίο ενδιαφέροντος, προσφέροντας πιο ισχυρό συνακτικό από τις RDF και RDF Schema καθώς και πιο ισχυρή σημασιολογία που βασίζεται στη λογική (logic–based semantics).

3.2 Οντολογίες

Οι οντολογίες [15], όπως αναφέρθηκε στην προηγούμενη ενότητα, αποτελούν το δομικό στοιχείο του Σημασιολογικού Ιστού. Ωστόσο, χρησιμοποιούνται ευρέως και στον τομέα της Τεχνητής Νοημοσύνης (Artificial Intelligence), σε εφαρμογές που σχετίζονται με τη διαχείριση της γνώσης, στο ηλεκτρονικό εμπόριο, στην ανάκτηση πληροφοριών, στην επεξεργασία της φυσικής γλώσσας και σε πολλούς ακόμα τομείς. Στην ενότητα αυτή περιγράφονται τα βασικότερα χαρακτηριστικά των οντολογιών.

3.2.1 Ορισμός της οντολογίας

Η οντολογία είναι μια έννοια που χρησιμοποιήθηκε για πρώτη φορά από τους αρχαίους Έλληνες φιλοσόφους στην προσπάθειά τους να απαντήσουν σε κάποια φιλοσοφικά ερωτήματα σχετικά με την ουσία και την ύπαρξη κάποιων πραγμάτων και εννοιών.

Με τον όρο οντολογία εννοούμε την ακριβή περιγραφή πραγμάτων και εννοιών καθώς και των σχέσεων που υπάρχουν ανάμεσα τους.

Ο πιο γνωστός ορισμός για την οντολογία, στην επιστήμη των υπολογιστών, πάνω στον οποίο στηρίχτηκαν και άλλοι ορισμοί, δόθηκε από τον Gruber [16] και είναι ο ακόλουθος:

- *An ontology is an explicit specification of a conceptualization.*

Παρακάτω αναφέρονται μερικοί ακόμα ενδεικτικοί ορισμοί που έχουν δοθεί για να περιγράψουν την έννοια της οντολογίας:

- *A logical theory which gives an explicit, partial account of a conceptualization* [43].

- *A set of logical axioms designed to account for the intended meaning of a vocabulary* [17].

- *An ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base* [18].

Οι οντολογίες μπορούν να χωριστούν σε δύο κατηγορίες. Η μια κατηγορία περιλαμβάνει οντολογίες που αποτελούν απλές ταξινομήσεις, και ονομάζονται *lightweight* οντολογίες. Στη δεύτερη κατηγορία ανήκουν οι οντολογίες οι οποίες μοντελοποιούν έννοιες και τις μεταξύ τους σχέσεις με τη χρήση αξιωμάτων και περιορισμών. Οι οντολογίες που ανήκουν σ'αυτή την κατηγορία ονομάζονται *heavyweight* οντολογίες.

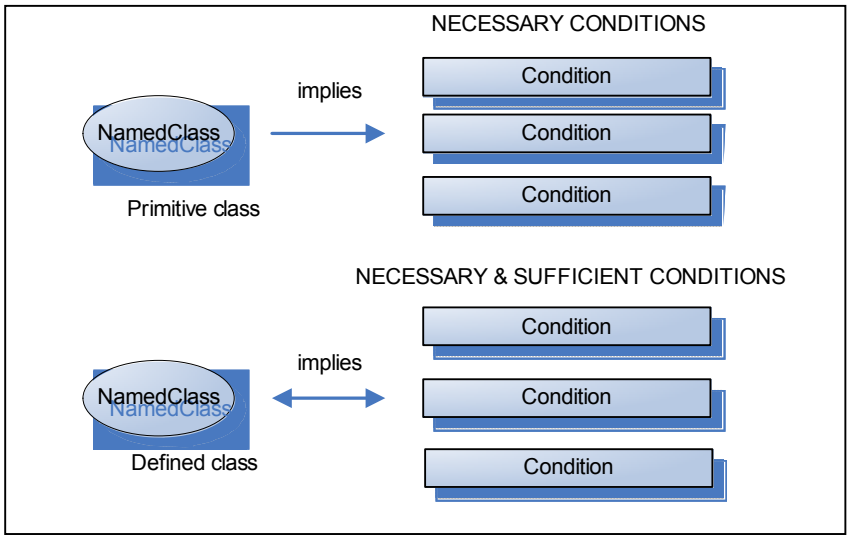
3.2.2 Τα κύρια συστατικά των οντολογιών

Στην ενότητα αυτή περιγράφονται τα κύρια συστατικά μιας οντολογίας με χρήση των Description Logics (DL). Ο όρος Description Logic αναφέρεται σε ένα υποσύνολο της λογικής πρώτης τάξης (First Order Logic), στο οποίο στηρίχθηκαν οι γλώσσες του Σημασιολογικού Ιστού, όπως η OWL, που δεν υποστηρίζει την ύπαρξη ελεύθερων μεταβλητών.

Μια DL οντολογία [44] αποτελείται από τρία είδη συστατικών: ***κλάσεις*** (*classes* ή *concepts*), ***σχέσεις*** (*roles* ή *properties*) και ***στιγμιότυπα*** (*individuals* ή *instances*).

Οι *κλάσεις* αναπαριστούν έννοιες, είτε αφηρημένες ή συγκεκριμένες. Οι κλάσεις είναι σύνολα από στιγμιότυπα και συνήθως είναι οργανωμένες σε μια ιεραρχία, η οποία είναι γνωστή και ως ταξινομία (taxonomy). Μπορούμε για παράδειγμα να αναπαραστήσουμε μια ταξινομία από τους φοιτητές ενός πανεπιστημίου (προπτυχιακοί, μεταπτυχιακοί, υποψήφιοι διδάκτορες κτλ). Σε αυτήν την ταξινομία η κλάση «Προπτυχιακός Φοιτητής» είναι υποκλάση της κλάσης «Φοιτητής». Οι κλάσεις στις DL οντολογίες διακρίνονται στις ακόλουθες κατηγορίες (εικόνα 3.1):

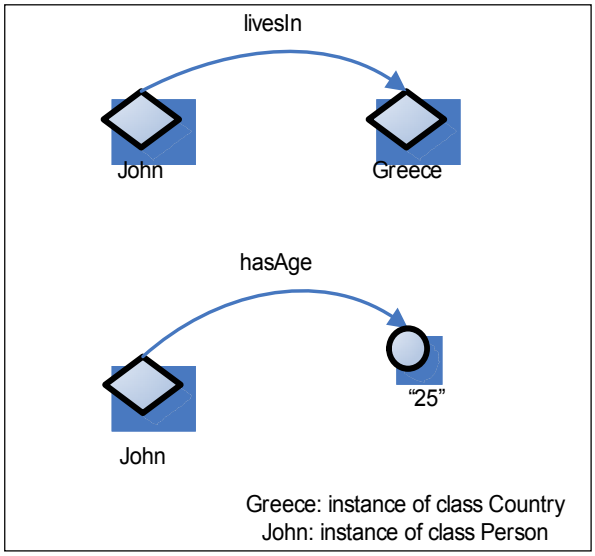
- Primitive: Στην κατηγορία αυτή ανήκουν οι κλάσεις οι οποίες περιγράφονται μόνο από αναγκαίες συνθήκες (necessary conditions). Αυτό σημαίνει πως αν κάποιο αντικείμενο είναι μέλος μιας κλάσης, τότε είναι αναγκαίο να ικανοποιεί τις αντίστοιχες συνθήκες.
- Defined: Στην κατηγορία αυτή ανήκουν οι κλάσεις οι οποίες περιγράφονται από ικανές και αναγκαίες συνθήκες (necessary and sufficient conditions). Αυτό σημαίνει ότι αν κάποιο αντικείμενο είναι μέλος μιας κλάσης, τότε είναι αναγκαίο να ικανοποιεί τις αντίστοιχες συνθήκες και αν ένα αντικείμενο ικανοποιεί τις συνθήκες τότε πρέπει να ανήκει στην αντίστοιχη κλάση.



Εικόνα 3.1 Ικανές και αναγκαίες συνθήκες

Οι *σχέσεις* αναπαριστούν δυαδικές συσχετίσεις ανάμεσα στα στιγμιότυπα των κλάσεων. Οι σχέσεις μπορεί να έχουν ένα πεδίο ορισμού (*domain*) και ένα πεδίο τιμών (*range*) ενώ συνδέουν στιγμιότυπα από το πεδίο ορισμού με στιγμιότυπα από το πεδίο τιμών. Το πεδίο τιμών μπορεί να είναι είτε μια κλάση ή ένας τύπος δεδομένων. Ανάλογα με το είδος του πεδίου τιμών οι σχέσεις διακρίνονται στις ακόλουθες κατηγορίες:

- Relations: Στην κατηγορία αυτή ανήκουν οι σχέσεις που συνδέουν ένα στιγμιότυπο με ένα άλλο στιγμιότυπο.
- Attributes: Στην κατηγορία αυτή ανήκουν οι σχέσεις που συνδέουν ένα στιγμιότυπο με έναν literal τύπο δεδομένων (π.χ. αριθμητικό, συμβολοσειρά, κτλ).



Εικόνα 3.2 Διαφορετικά είδη σχέσεων

Στην εικόνα 3.2, για παράδειγμα, η κλάση “Person” είναι το domain της σχέσης “livesIn”, ενώ η κλάση “Country” αποτελεί το range της ίδιας σχέσης.

Όπως και οι κλάσεις, έτσι και οι σχέσεις είναι δυνατόν να οργανωθούν σε ιεραρχίες. Για παράδειγμα η σχέση “hasMother” αποτελεί υπο-σχέση (sub-property) της σχέσης “hasParent”.

Οι στοιχειώδεις DL οντολογίες περιέχουν ατομικές κλάσεις και ατομικές σχέσεις. Είναι όμως δυνατό να οριστούν σύνθετες κλάσεις με τη βοήθεια των DL constructs. (πίνακας 3.1). Επίσης, οι DLs προσφέρουν αξιώματα (πίνακας 3.2) τα οποία περιγράφουν τον τρόπο με τον οποίο οι κλάσεις ή οι σχέσεις σχετίζονται μεταξύ τους.

Πίνακας 3.1 DL constructs (C, D: concepts – R: property)

OWL σύνταξη	DL	Παράδειγμα	Περιγραφή
IntersectionOf	C ∩ D	Worker ∩ Male	All Workers that are Male
unionOf	C ∪ D	Worker ∪ Male	Anyone that is either a Worker or Male
Atomic megation	¬ C	¬ Male	Any individual that is not Male
allValuesFrom	∀R.C	∀manager.Male	All managers must be of type Male
someValuesFrom	∃R.C	∃hasChild.Male	At least one of the children must be of type Male
Value	∃R.{o}	∃hasLocation.*Athens*	The location property must have the value Athens
minCardinality	≥n R.C	≥1 occupies.Worker	A Department occupies at least one Worker
maxCardinality	≤n R.C	≤1 worksIn.Department	A Worker works at most in a Department
Cardinality	=n R.C	=1 livesIn.Country	A Person lives exactly in one Country

Πίνακας 3.2 DL αξιώματα (μπορούν να εφαρμοστούν και σε σχέσεις) (C, D: concepts or roles)

OWL ορολογία	DL	Παράδειγμα	Περιγραφή
Inclusion (subsumption)	C ⊆ D	Worker ⊆ Person	Any individual of type Worker is also of type Person
Equality	C ≡ D	Young ≡ Teenager	Every Young is also a Teenager and vice versa
Disjoint	C ¬ D	Male ¬ Female	Someone cannot be a Male and Female at the same time

Τα ***στιγμιότυπα*** αποτελούν διακριτά αντικείμενα-μέλη των κλάσεων. Για παράδειγμα στην εικόνα 3.2, οι “John” και “Greece” αποτελούν στιγμιότυπα των κλάσεων “Person” και ”Country” αντίστοιχα.

3.2.3 Η γλώσσα RDF και RDFS

Η RDF [38] (Resource Description Framework) είναι ένα W3C πρότυπο με το οποίο περιγράφονται μεταδεδομένα στο δίκτυο. Η RDF προσφέρει ένα μοντέλο δεδομένων για την περιγραφή πληροφοριών έτσι ώστε να είναι δυνατή η ανάγνωση και η κατανόησή τους από τους υπολογιστές.

Το RDF μοντέλο δεδομένων αποτελείται από τρία συστατικά:

- *Resources:* Μπορούμε να θεωρήσουμε ένα resource ως ένα αντικείμενο, ένα πράγμα για το οποίο θέλουμε να μιλήσουμε (π.χ. βιβλίο, συγγραφέας, σπίτι, κτλ). Η αναφορά σε ένα resource γίνεται με τη χρήση ενός URI (Universal Resource Identifier) το οποίο μπορεί να είναι ένα URL (Unified Resource Locator) ή οτιδήποτε άλλο μπορεί να προσδιορίσει μοναδικά ένα resource.
- *Properties:* Ορίζουν ιδιότητες και σχέσεις με τις οποίες περιγράφονται τα resources (π.χ. ηλικία, τίτλος, κατάγεται από, κτλ). Και τα properties αναγνωρίζονται με τη χρήση URIs.
- *Statements:* Εκχωρούν μια τιμή σε ένα property για ένα συγκεκριμένο resource. Υπάρχουν τρεις τρόποι αναπαράστασης των RDF statements: α) με τη χρήση τριπλέτων, β) με τη χρήση κατευθυνόμενων γράφων με ετικέτες στις ακμές (directed labeled graphs) και γ) με τη χρήση ενός συντακτικού παρόμοιου με την XML (XML-like syntax). Στην αναπαράσταση με τη χρήση τριπλέτων, ένα RDF statement αποτελείται από τρία επιμέρους συστατικά: subject, property (ή predicate) και object. Παράδειγμα ενός statement αποτελεί η πρόταση: “Dan Brown is the owner of the Web page of http://www.danbrown.com”, στο οποίο: subject = “Dan Brown”, property = “owner” και object = “http://www.danbrown.com”. Εναλλακτικά, τα RDF statements μπορούν να μοντελοποιηθούν με κόμβους και ακμές σε έναν γράφο. Στην περίπτωση αυτή ένα statement αναπαρίσταται από έναν κόμβο για το subject, έναν για το object και μια ακμή για το predicate με κατεύθυνση από το subject στο object. Η αναπαράσταση με τη χρήση ενός XML-like συντακτικού χρησιμοποιείται ώστε να είναι δυνατή η επεξεργασία των RDF statements από τους υπολογιστές.

Το RDF μοντέλο δεδομένων δεν προσφέρει τη δυνατότητα δήλωσης περιορισμών στις σχέσεις που υπάρχουν ανάμεσα στα properties και τα resources. Για παράδειγμα, στην RDF γλώσσα δεν μπορούμε να ορίσουμε ότι η σχέση “livesIn” της εικόνας 3.2 υφίσταται

μόνο ανάμεσα στα αντικείμενα των κλάσεων “Person” και “Country”. Ο περιορισμός αυτός αντιμετωπίζεται με τη χρήση μιας επέκτασης του RDF μοντέλου, που λέγεται RDF Schema (RDFS). Με την RDFS είναι δυνατός ο ορισμός των resources ως αντικείμενα κλάσεων. Ο συνδυασμός των RDF και RDFS είναι γνωστός ως RDF(S). Με την RDF(S) είναι δυνατή η περιγραφή οντολογιών που αποτελούν απλές ταξινομίες. Πιο συγκεκριμένα η RDF(S) γλώσσα δεν προσφέρει τη δυνατότητα περιγραφής σύνθετων οντολογιών, δηλαδή οντολογιών που περιέχουν περιορισμούς.

Στην εικόνα 3.3 δίνεται ένα παράδειγμα χρήσης της RDF(S) για τον ορισμό της σχέσης “livesIn” της εικόνας 3.2.

```
<rdfs:Class rdf:ID=“Person”>
   <rdfs:comment>A class that contains all the people</rdfs:comment>
</rdfs:Class>
<rdfs:Class rdf:ID=”Country”>
  <rdfs:comment>A class that contains all the countries</rdfs:comment>
</rdfs:Class>
<rdf:Property rdf:ID=“livesIn”>
   <rdfs:domain rdf:resource=”#Person”/>
   <rdfs:range rdf:resource=”#Country”/>
</rdf:Property>
```

Εικόνα 3.3 Περιγραφή της σχέσης “livesIn” σε RDF(S) (τα στοιχεία της γλώσσας εμφανίζονται με έντονα γράμματα).

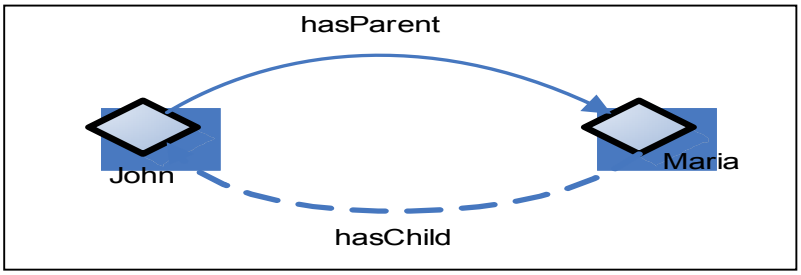
3.2.4 Η γλώσσα OWL

Η OWL [45] (Web Ontology Language) είναι μια πολύ δημοφιλή γλώσσα περιγραφής οντολογιών, πιο εκφραστική ως προς την αναπαράσταση του νοήματος και της σημασιολογίας κάποιων όρων και σχέσεων από τις RDF και RDF(S). Αποτελεί την πιο πρόσφατα ανεπτυγμένη γλώσσα για οντολογίες από το W3C. Το πιο συνηθισμένο συντακτικό που ακολουθεί η OWL βασίζεται στην RDF/XML. Η OWL μπορεί να κατηγοριοποιηθεί σε τρεις υπο-γλώσσες (sub-languages):

- OWL-Lite: Είναι η πιο απλή συντακτικά και η λιγότερο εκφραστική υπο-γλώσσα. Χρησιμοποιείται για οντολογίες που αποτελούνται από μια απλή ιεραρχία κλάσεων και απλούς περιορισμούς.

- OWL-DL: Είναι πιο εκφραστική από την OWL-Lite και βασίζεται στα Description Logics. Μια οντολογία εκφρασμένη σε OWL-DL μπορεί να χρησιμοποιηθεί από κάποιο μηχανισμό εξαγωγής συμπερασμάτων καθώς και να ελεγχθεί αυτόματα αν περιέχει ασυνέπειες.
- OWL-Full: Είναι η πιο εκφραστική υπο-γλώσσα της OWL. Χρησιμοποιείται σε περιπτώσεις που απαιτείται μεγάλη εκφραστικότητα, ενώ δεν είναι δυνατόν να γίνει αυτόματος συμπερασμός σε μια OWL-Full οντολογία.

Μια OWL οντολογία αποτελείται από κλάσεις (concepts), σχέσεις (properties) και στιγμιότυπα (individuals) όπως αυτά περιγράφτηκαν στην παράγραφο 3.2.2. Η OWL επιτρέπει την οργάνωση των κλάσεων και των σχέσεων σε μια ιεραρχία (subclass/superclass και subproperty/superproperty αντίστοιχα). Όταν και το πεδίο ορισμού και το πεδίο τιμών μιας σχέσης αποτελείται από κλάσεις (primitive ή defined), η σχέση καλείται ObjectProperty ενώ αν το πεδίο τιμών μιας σχέσης είναι ένας τύπος δεδομένων (XML schema datatype - XSD) η σχέση είναι γνωστή ως DatatypeProperty.



Εικόνα 3.4 Παράδειγμα αντίστροφης σχέσης.

Στην OWL, οι σχέσεις μπορούν να διέπονται από αξιώματα [44]. Έτσι λοιπόν μια σχέση μπορεί να είναι:

- *Αντίστροφη (inverse property):* Εάν μια σχέση s συνδέει το στιγμιότυπο a με το στιγμιότυπο b τότε η αντίστροφη σχέση της s θα συνδέει το στιγμιότυπο b με το στιγμιότυπο a. Η εικόνα 3.4 δίνει ένα παράδειγμα μιας αντίστροφης σχέσης. Στην εικόνα αυτή η σχέση “hasChild” είναι αντίστροφη σχέση της σχέσης “hasParent”.
- *Συναρτησιακή (functional property):* Αν μια σχέση είναι συναρτησιακή για ένα στιγμιότυπο a, τότε το πολύ ένα στιγμιότυπο μπορεί να σχετίζεται με το στιγμιότυπο a μέσω της συγκεκριμένης σχέσης. Ένα παράδειγμα συναρτησιακής σχέσης αποτελεί η σχέση “hasMother”, αφού οποιοσδήποτε άνθρωπος έχει μια μόνο μητέρα.

- *Αντίστροφα συναρτησιακή (inverse functional property):* Αν μια σχέση s είναι αντίστροφα συναρτησιακή τότε η αντίστροφη σχέση της s είναι συναρτησιακή. Ένα παράδειγμα αντίστροφα συναρτησιακής σχέσης είναι η σχέση “isMotherOf”, αφού η αντίστροφη σχέση “hasMother” είναι συναρτησιακή.

- *Μεταβατική (transitive property):* Αν μια σχέση s είναι μεταβατική τότε αν η s συνδέει το στιγμιότυπο a με το στιγμιότυπο b και το στιγμιότυπο b με το στιγμιότυπο c, τότε και το στιγμιότυπο a συνδέεται με το στιγμιότυπο c μέσω της σχέσης s. Μεταβατική μπορεί να είναι η σχέση “hasAncestor”, αφού αν ο Α είναι πρόγονος του Β και ο Β πρόγονος του Γ, τότε ο Α είναι πρόγονος του Γ.

- *Συμμετρική (symmetric property):* Αν μια σχέση s είναι συμμετρική και η s συνδέει το στιγμιότυπο a με το στιγμιότυπο b, τότε και το στιγμιότυπο b συνδέεται με το στιγμιότυπο a μέσω της s. Ένα παράδειγμα συμμετρικής σχέσης είναι η σχέση “hasCousin”, αφού αν ο Α έχει ξάδερφο τον Β, τότε και ο Β έχει ξάδερφο τον Α.

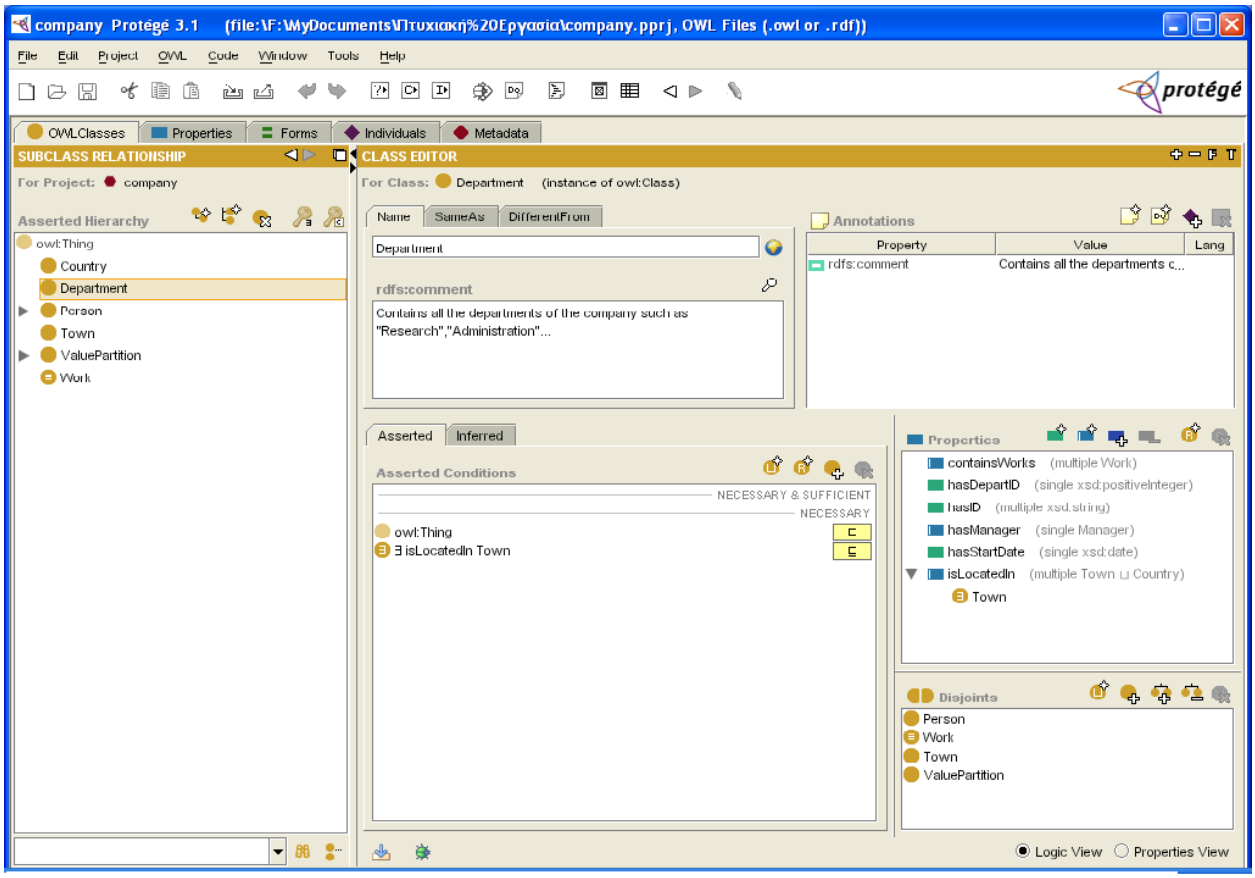
Στην εικόνα 3.5, δίνεται σε OWL ο ορισμός της σχέσης “hasManager”, η οποία συνδέει ένα στιγμιότυπο από την κλάση “Department” με ένα στιγμιότυπο από την κλάση “Manager”. Επειδή κάθε στιγμιότυπο της κλάσης “Department” συνδέεται το πολύ με ένα στιγμιότυπο της κλάσης “Manager” μέσω της σχέσης “hasManager”, η αντίστροφη κλάση “isManagerOf” είναι αντίστροφα συναρτησιακή.

```
<owl:ObjectProperty rdf:ID="hasManager">
  <rdfs:domain rdf:resource="#Department"/>
  <owl:inverseOf>
    <owl:InverseFunctionalProperty rdf:about="#isManagerOf"/>
  </owl:inverseOf>
  <rdfs:range rdf:resource="#Manager"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
 </owl:ObjectProperty>
```

Εικόνα 3.5 Παράδειγμα χρήσης της OWL για τον ορισμό μιας σχέσης σε XML σύνταξη (τα στοιχεία της γλώσσας εμφανίζονται με έντονα γράμματα).

3.2.5 Εργαλεία ανάπτυξης οντολογιών

Εκτός από γλώσσες ορισμού οντολογιών υπάρχουν και εργαλεία με γραφικό περιβάλλον που διευκολύνουν τη δημιουργία οντολογιών. Μερικά από αυτά είναι: Protégé [21], WebODE [46], OntoEdit [47], κ.ά. Στην παρούσα εργασία χρησιμοποιήθηκε το Protégé 3.1 το οποίο βασίζεται στη γλώσσα Java [48] και κατασκευάστηκε στο πανεπιστήμιο του Stanford. Στην εικόνα 3.6 φαίνεται ένα στιγμιότυπο του Protégé που αναπαριστά τις κλάσεις μιας οντολογίας.



Εικόνα 3.6 Στιγμιότυπο του Protégé που αναπαριστά τις κλάσεις μιας οντολογίας.

ΚΕΦΑΛΑΙΟ 4

ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Το ταίριασμα ενός μοντέλου δεδομένων με κάποιο άλλο (schema matching), όπως είναι και η απεικόνιση του σχεσιακού μοντέλου σε οντολογία, είναι ένα πεδίο έρευνας το οποίο προσελκύει το ενδιαφέρον πολλών ερευνητών από την προηγούμενη δεκαετία. Μερικές από τις εφαρμογές του είναι το ηλεκτρονικό εμπόριο, η αποθήκευση δεδομένων (data warehousing) καθώς και η σημασιολογική επεξεργασία επερωτήσεων. Κατά καιρούς έχουν προταθεί κάποιες τεχνικές και αλγόριθμοι που στόχο έχουν την αντιστοίχιση ενός μοντέλου δεδομένων σε κάποιο άλλο. Μια υποκατηγορία αυτών των τεχνικών αφορά στην αντιστοίχιση ενός σχεσιακού σχήματος βάσεων δεδομένων σε μια οντολογία.

Στην παρούσα ενότητα παρουσιάζονται κάποια εργαλεία και γλώσσες που αντιστοιχίζουν ένα σχεσιακό σχήμα βάσης δεδομένων σε μια οντολογία. Στο τέλος του κεφαλαίου συνοψίζονται τα πιο σημαντικά χαρακτηριστικά των συστημάτων αυτών σε έναν πίνακα και περιγράφονται οι προδιαγραφές ενός συστήματος ταιριάσματος δύο μοντέλων δεδομένων.

4.1 KAON Reverse

Το KAON Reverse [7,8] είναι ένα ημι-αυτόματο εργαλείο, με γραφικό για το χρήστη περιβάλλον (εικόνα 4.2), το οποίο κατασκευάστηκε από το Πανεπιστήμιο της Karlsruhe της Γερμανίας με σκοπό τη μεταφορά της πληροφορίας που είναι αποθηκευμένη σε μια βάση δεδομένων σε μια οντολογία. Η οντολογία δεν δημιουργείται αυτόματα από τη βάση δεδομένων, αλλά πρέπει να υπάρχει πριν αρχίσει η διαδικασία της αντιστοίχισης.

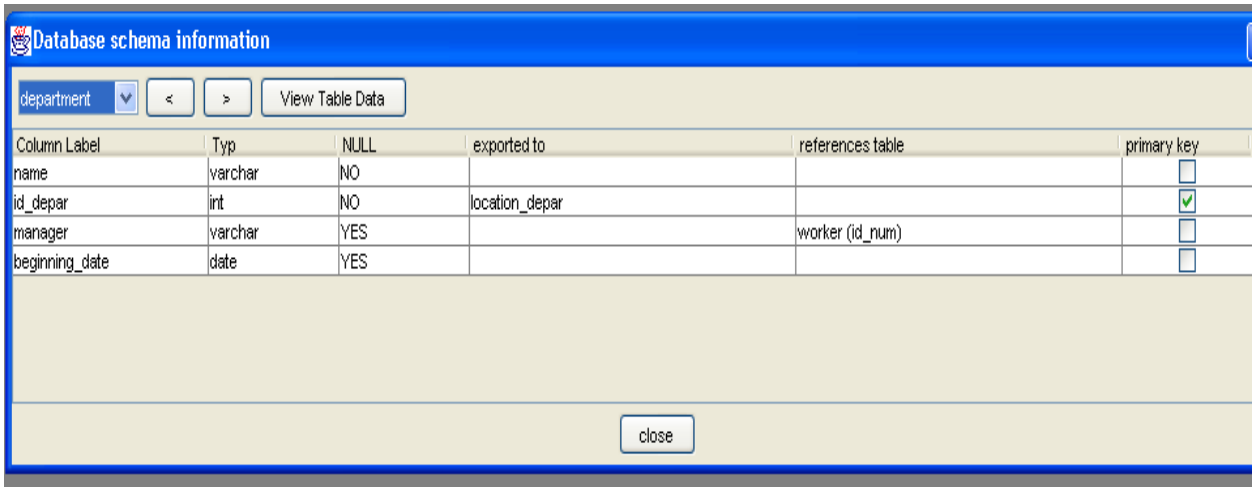
Η διαδικασία της αντιστοίχισης που ακολουθείται από το συγκεκριμένο εργαλείο περιλαμβάνει τα παρακάτω βήματα.

1. Εξαγωγή των μετα-δεδομένων (σχέσεις, γνωρίσματα, πρωτεύοντα κλειδιά, ξένα κλειδιά κτλ) από το σχεσιακό σχήμα με χρήση μιας τεχνικής *reverse engineering*.
2. Ανάλυση της πληροφορίας που αντλήθηκε στο προηγούμενο βήμα για την αντιστοίχιση των στοιχείων της βάσης στα στοιχεία της οντολογίας με τη βοήθεια κανόνων αντιστοίχισης. Κάποιοι από τους κανόνες αυτούς είναι και οι εξής:
   2.1 Κάθε σχέση της βάσης δεδομένων αντιστοιχίζεται σε μια κλάση της οντολογίας. Τον κανόνα αυτό δεν ακολουθούν οι πίνακες που εκφράζουν n:m σχέσεις ανάμεσα σε δύο άλλους πίνακες καθώς και οι πίνακες οι οποίοι πρέπει να συνδυαστούν προκειμένου να αντιστοιχηθούν σε κάποια κλάση.
   2.2 Τα γνωρίσματα των πινάκων της βάσης δεδομένων μετατρέπονται σε properties της οντολογίας.
   2.3 Για κάθε εξάρτηση εγκλεισμού (inclusion dependency) που υπάρχει στους πίνακες δημιουργείται μια σχέση κληρονομικότητας (inheritance relationship).
3. Έλεγχος αν έχουν αντιστοιχηθεί όλα τα στοιχεία του σχεσιακού σχήματος στα στοιχεία της οντολογίας.
4. Μεταφορά των δεδομένων από τη βάση στην οντολογία σύμφωνα με τους κανόνες αντιστοίχισης που ορίστηκαν στα προηγούμενα βήματα.

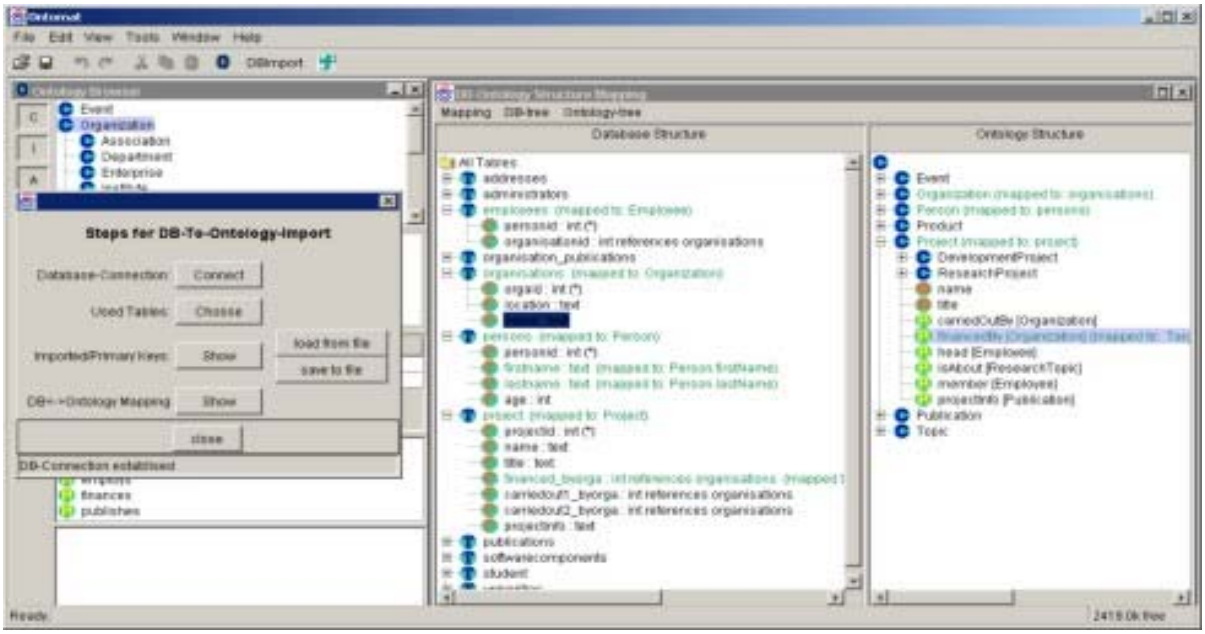
Η διαδικασία της αντιστοίχισης χωρίζεται σε δύο κύρια μέρη: στην αντιστοίχιση πινάκων (Table Mapping) και στην αντιστοίχιση στηλών (Column Mapping). Πιο συγκεκριμένα, οι πίνακες του σχεσιακού σχήματος της βάσης δεδομένων αντιστοιχίζονται σε κλάσεις της οντολογίας και οι στήλες των πινάκων αυτών σε datatype properties (λέγονται και attributes) ή σε object properties (λέγονται και relations) της οντολογίας. Το KAON Reverse δίνει στους χρήστες τη δυνατότητα να επιλέξουν ανάμεσα σε απλή ή σύνθετη αντιστοίχιση. Στην πρώτη περίπτωση όλες οι πλειάδες του πίνακα «μετατρέπονται» σε στιγμιότυπα (instances) της οντολογίας, ενώ στη δεύτερη περίπτωση η δημιουργία των στιγμιοτύπων περιορίζεται από SQL επερωτήσεις. Παρέχεται επίσης η δυνατότητα ρητής δήλωσης των μεταδεδομένων της βάσης - πρωτεύοντα και ξένα κλειδιά – σε περίπτωση που ο χρησιμοποιούμενος JDBC driver δεν προσφέρει μεθόδους για το σκοπό αυτό (εικόνα 4.1).

Αξίζει να σημειωθεί ότι το KAON Reverse προσφέρει μεθόδους εύρεσης της λεξικογραφικής ομοιότητας προκειμένου να ανακαλύψει πιθανή αντιστοίχιση ανάμεσα στα στοιχεία των σχημάτων.

Η οντολογία που χρησιμοποιείται από το συγκεκριμένο εργαλείο είναι υλοποιημένη σε RDF(S) και ως εκ τούτου δεν προσφέρει δυνατότητα αναπαράστασης σύνθετων οντολογιών. Συγκεκριμένα, το KAON Reverse δεν υποστηρίζει οντολογίες που περιέχουν αξιώματα (axiomatized ontologies) αλλά και relations που το domain τους είναι ένα σύνολο από κλάσεις.



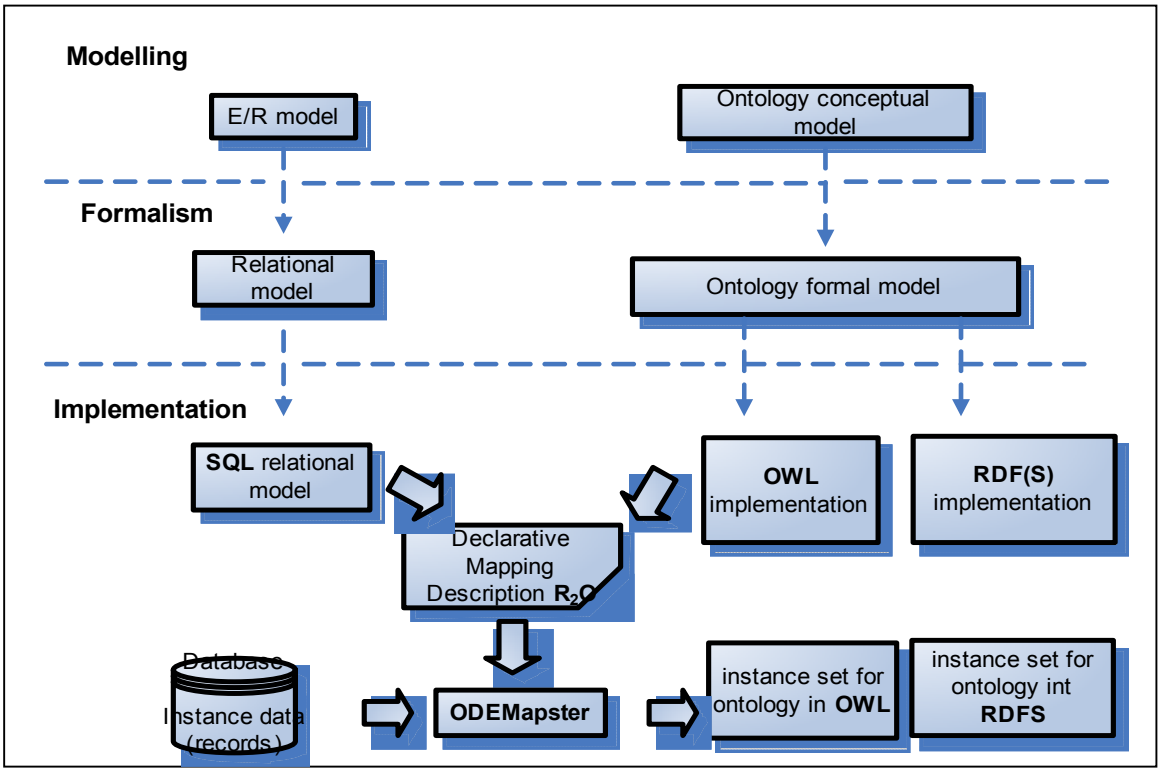
Εικόνα 4.1 KAON Reverse: Εισαγωγή των μεταδεδομένων (πρωτεύοντα και ξένα κλειδιά) από το χρήστη.



Εικόνα 4.2 Χαρακτηριστική οθόνη του KAON Reverse. Στο δεξί μέρος διακρίνονται τα προς αντιστοίχιση σχήματα.

4.2 R_2O (Relational to Ontology)

Η R_2O [23] είναι μια δηλωτική γλώσσα η οποία περιγράφει αντιστοιχίσεις ανάμεσα σε σχεσιακά σχήματα βάσεων δεδομένων και οντολογίες υλοποιημένες σε RDF(S) ή OWL. Η προσέγγιση η οποία ακολουθείται από τη συγκεκριμένη γλώσσα περιλαμβάνει τον ορισμό των αντιστοιχίσεων ανάμεσα στα στοιχεία του σχεσιακού σχήματος και της οντολογίας, η οποία προϋπάρχει και δεν παράγεται κατά τη διαδικασία της αντιστοίχισης, και την αυτόματη επεξεργασία των αντιστοιχίσεων αυτών από τον ODEMapster επεξεργαστή αντιστοιχίσεων (εικόνα 4.3).



Εικόνα 4.3 Αρχιτεκτονική R_2O

Η R_2O είναι μια γλώσσα ανεξάρτητη των Σχεσιακών Συστημάτων Διαχείρισης Βάσεων Δεδομένων και μπορεί να χρησιμοποιηθεί για οποιαδήποτε βάση υλοποιεί το SQL πρότυπο. Είναι επίσης δυνατή η χρήση της από υπάρχοντα εργαλεία, που στόχο έχουν την απεικόνιση ενός σχεσιακού σχήματος σε μια οντολογία, για τη δήλωση αντιστοιχίσεων ανάμεσα στα δύο αυτά σχήματα.

Οι αντιστοιχίσεις που ορίζονται από τη γλώσσα R_2O χρησιμοποιούνται για την εξαγωγή των δεδομένων από τη βάση και τη «μεταφορά» τους στην οντολογία (data migration). Η R_2O δεν αποφασίζει το βαθμό ομοιότητας των στοιχείων της βάσης δεδομένων με αυτών της οντολογίας αλλά εκφράζει τις συνθήκες και τους μετασχηματισμούς κάτω από τους οποίους τα στοιχεία αυτά είναι όμοια. Ένα από τα σημαντικότερα

χαρακτηριστικά της γλώσσας είναι η υποστήριξη αντιστοιχίσεων μεταξύ ενός συνόλου από πίνακες, στους οποίους έχει εφαρμοστεί μία ή περισσότερες από τις πράξεις της σχεσιακής άλγεβρας, και μιας κλάσης της οντολογίας.

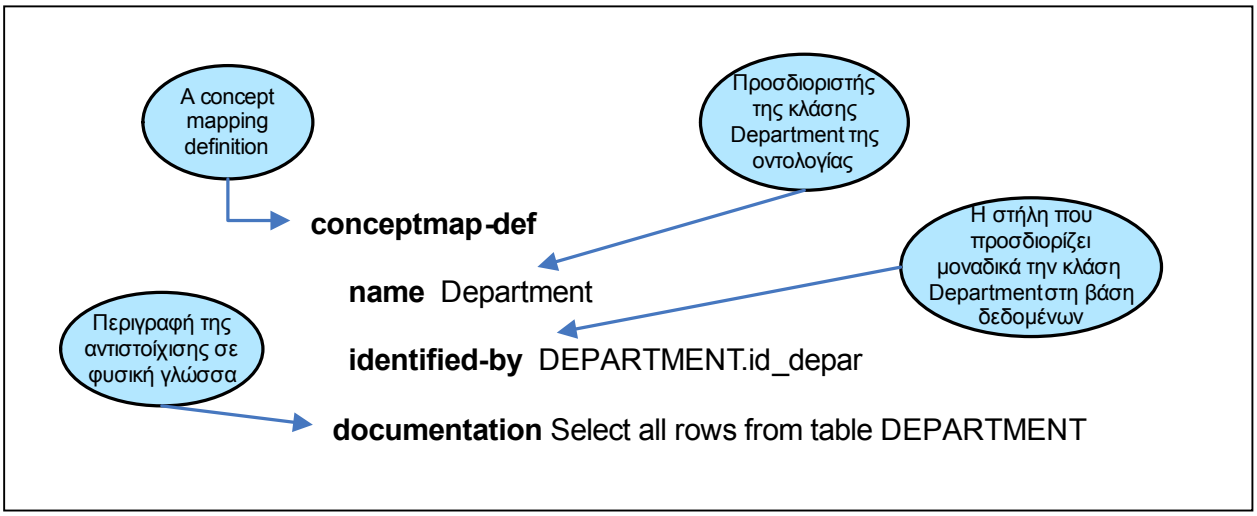
Το συντακτικό που χρησιμοποιεί η R_2O είναι σχεδιασμένο με βάση αυτό της XML. Περιέχει συστατικά τα οποία περιγράφουν:

- τη δομή των σχημάτων σχεσιακών βάσεων δεδομένων,
- τις αντιστοιχίσεις των κλάσεων (concept mappings),
- τις συνθήκες και τους μετασχηματισμούς που πρέπει να ληφθούν υπόψη για να γίνει μια αντιστοίχιση,
- τις αντιστοιχίσεις γνωρισμάτων και σχέσεων (attribute and relation mappings).

```
dbschema-desc
  name COMPANY
  has_table
    name Department
    documentation "Stores information about the departments of the company"
    keycol-desc
      name Department.id_depar
      columnType integer
      documentation "Identifies a department"
    nonkeycol-desc
      name Department.name
      columnType string
    nonkeycol-desc
      name Department.beginning_date
      columnType date
    forkeycol-desc
      name Department.manager
      columnType string
      refers-to Worker.id_num
      documentation "Points at a worker who is manager of the department"
```

Εικόνα 4.4 Παράδειγμα χρήσης της γλώσσας R_2O για την περιγραφή του σχεσιακού σχήματος.

Στην εικόνα 4.4 δίνεται ένα παράδειγμα χρήσης της γλώσσας για την περιγραφή της σχέσης DEPARTMENT(name, id_depar, manager, beginning_name) της βάσης δεδομένων COMPANY, ενώ στην εικόνα 4.5 δίνεται ένα παράδειγμα αντιστοίχισης της σχέσης αυτής στην κλάση Department της οντολογίας. Με έντονα γράμματα είναι σημειωμένα τα στοιχεία που χρησιμοποιεί η γλώσσα για την περιγραφή αυτή.

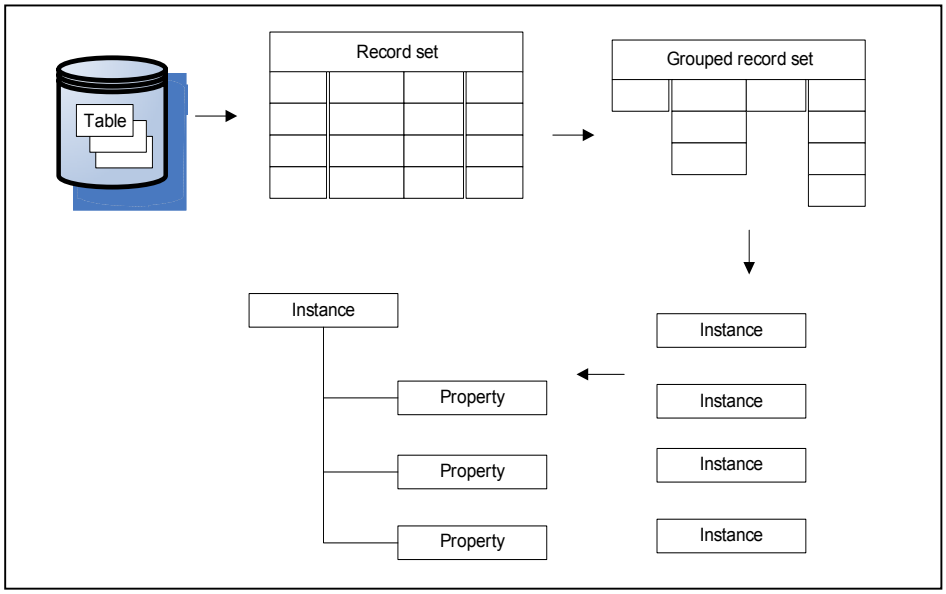


Εικόνα 4.5 Παράδειγμα χρήσης της R_2O για την αντιστοίχιση ενός πίνακα σε μια κλάση.

4.3 D2R Map (Database to RDF Mapping Language)

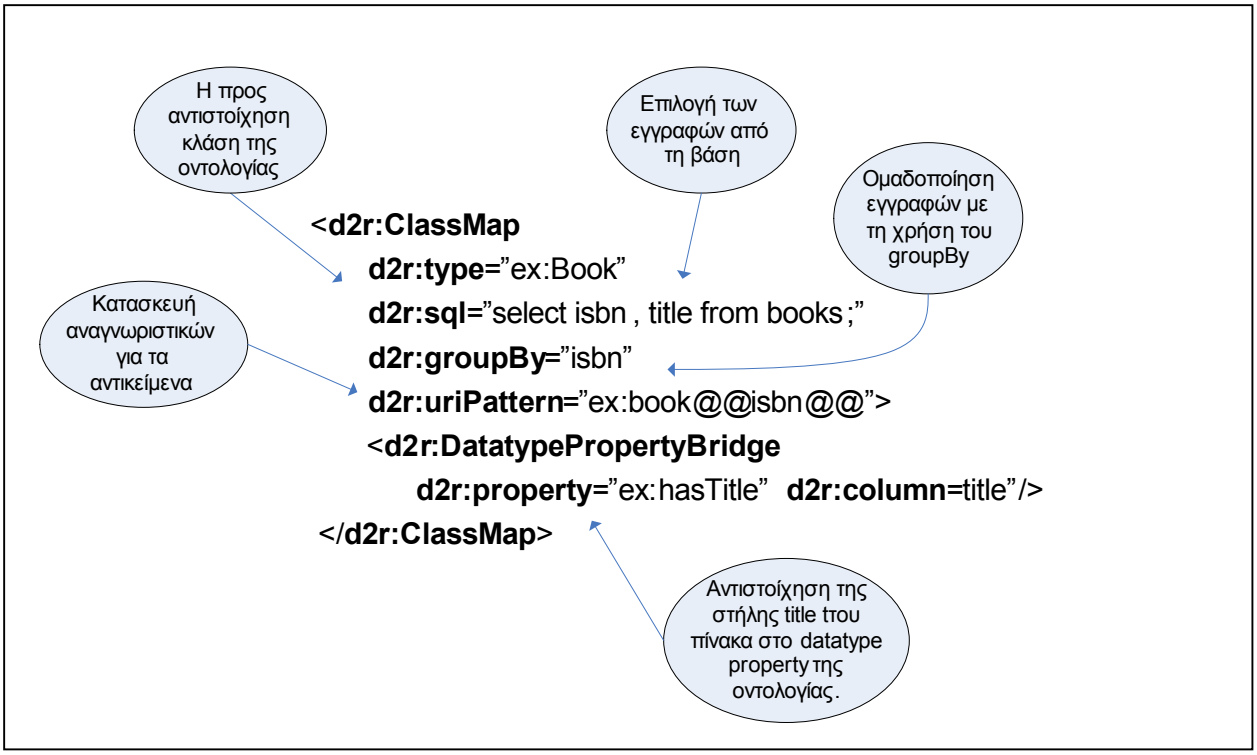
Η D2R [8] είναι μια δηλωτική γλώσσα, με συντακτικό βασισμένο στην XML, η οποία εκφράζει αντιστοιχίσεις ανάμεσα σε σχεσιακά σχήματα βάσεων δεδομένων και σε OWL οντολογίες. Οι αντιστοιχίσεις αυτές χρησιμοποιούνται από τον D2R επεξεργαστή για να εξάγουν δεδομένα από τη βάση σε RDF. Η διαδικασία αυτή πραγματοποιείται σε τέσσερα στάδια (εικόνα 4.6).

1. Επιλογή του συνόλου εγγραφών από τη βάση με τη χρήση της γλώσσας επερωτήσεων SQL για κάθε κλάση ή ομάδα από «όμοιες» κλάσεις.
2. Ομαδοποίηση των εγγραφών ανάλογα με την προς αντιστοίχιση κλάση.
3. Δημιουργία των αντικειμένων της κλάσης και κατασκευή προσδιοριστικών (identifiers).
4. Αντιστοίχιση του ομαδοποιημένου συνόλου εγγραφών σε αντικείμενα των properties.



Εικόνα 4.6 Η D2R διαδικασία αντιστοίχισης.

Στη συνέχεια δίνεται ένα παράδειγμα στο οποίο εφαρμόζεται η παραπάνω διαδικασία (εικόνα 4.7). Υποθέτουμε ότι έχουμε έναν πίνακα στη βάση ο οποίος αποθηκεύει πληροφορίες για όλα τα βιβλία μιας βιβλιοθήκης : Book (isbn, title) και θέλουμε να εκφράσουμε με τη βοήθεια της γλώσσας D2R την αντιστοιχία του πίνακα αυτού με την κλάση ex:Book της οντολογίας η οποία χαρακτηρίζεται από το datatype property ex:hasTitle.



Εικόνα 4.7 Παράδειγμα χρήσης της γλώσσας D2R. Τα στοιχεία της γλώσσας εμφανίζονται με έντονα γράμματα.

Σύμφωνα με τα βήματα που αναφέρθηκαν παραπάνω επιλέγονται, με τη χρήση της γλώσσας SQL, οι εγγραφές του πίνακα Book (δηλαδή όλα τα βιβλία) και ομαδοποιούνται με βάση το πεδίο isbn. Στη συνέχεια δημιουργείται ένα στιγμιότυπο για κάθε μία από τις εγγραφές τα οποία θα «εμπλουτίσουν» την κλάση ex:Book. Τέλος, στο datatype property ex:title, που σχετίζεται με την κλάση, αντιστοιχίζεται η στήλη title του συνόλου εγγραφών.

Η D2R προσφέρει ένα σύνολο από συστατικά τα οποία βοηθούν στο να περιγραφούν οι ακόλουθες λειτουργίες:

- Σύνδεση στη βάση δεδομένων (*d2r:DBConnection*, *d2r:username*, κτλ).
- Προσθήκη κειμένου στην αρχή ή στο τέλος του αρχείου εξόδου (*d2r:Prepend*, *d2r:Postpend*).
- Αποστολή μηνυμάτων στο D2R επεξεργαστή (*d2r:ProcessorMessage*).
- Αντιστοίχιση της πληροφορίας της βάσης στην οντολογία (*d2r:ClassMap*, *d2r:ObjectPropertyBridge*, *d2r:DatatypePropertyBridge*).
- Χρήση προτύπων στις τιμές των στηλών των πινάκων προτού οι τιμές αυτές χρησιμοποιηθούν σαν αντικείμενα των στοιχείων της οντολογίας (*d2r:pattern*).
- Χρήση πινάκων που στόχο έχουν τον μετασχηματισμό των στηλών προτού οι τιμές που φέρουν χρησιμοποιηθούν σαν αντικείμενα των στοιχείων της οντολογίας (*d2r:TranslationTable*, *d2r:Translation*)

Αξίζει να σημειωθεί ότι η D2R γλώσσα προσφέρει λύσεις για την αντιστοίχιση σχέσεων της βάσης δεδομένων που αναπαριστούν ν-αδικές συσχετίσεις. Διαχειρίζεται επίσης περιπτώσεις στις οποίες τα δεδομένα που αντιστοιχούν σε μια κλάση κατανέμονται σε περισσότερους του ενός πίνακες.

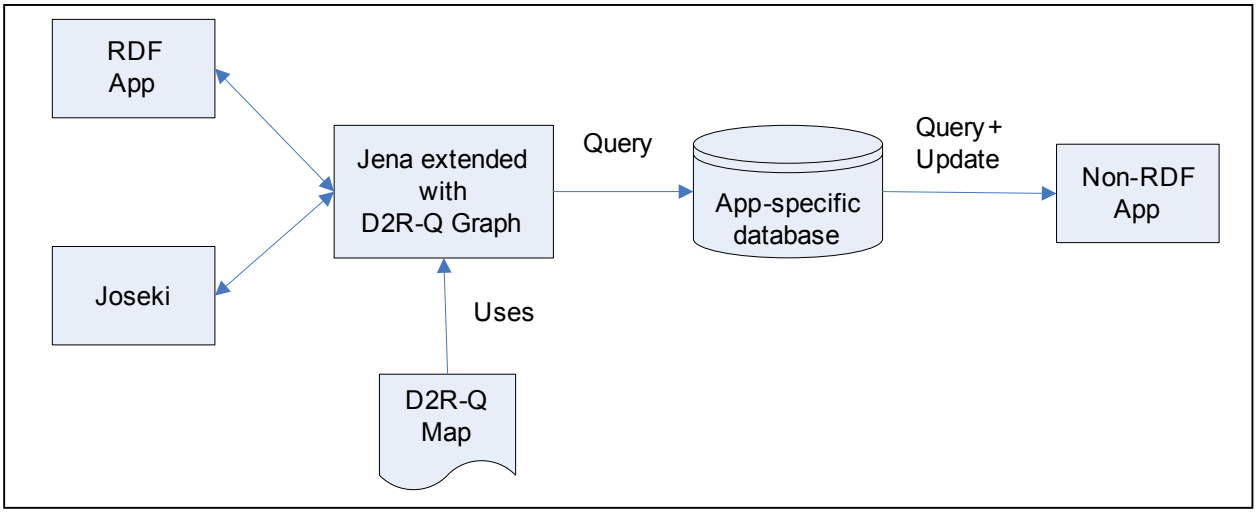
4.4 D2RQ

Η D2RQ [11] είναι μια δηλωτική, βασισμένη στην D2R Map, γλώσσα η οποία χρησιμοποιείται για την περιγραφή αντιστοιχίσεων ανάμεσα σε σχεσιακά σχήματα βάσεων δεδομένων και RDFS/OWL οντολογίες. Το συντακτικό της είναι βασισμένο στην XML γλώσσα και χρησιμοποιεί σχεδόν τα ίδια στοιχεία με αυτά της γλώσσας D2R (*d2rq:ClassMap*, *d2rq:DatatypePropertyBridge*, *d2rq:ObjectPropertyBridge*, κτλ)

Η D2RQ (εικόνα 4.8) αποτελείται από:

- την D2RQ Mapping Language (D2RQ Map), μια δηλωτική γλώσσα που περιγράφει τις αντιστοιχίσεις ανάμεσα στα στοιχεία της οντολογίας και του σχεσιακού μοντέλου δεδομένων και
- το GraphD2RQ, ένα plug-in για το εργαλείο Jena[1] [4] το οποίο διαχειρίζεται τις αντιστοιχίσεις που έχουν περιγραφεί από την D2RQ Map.

Οι αντιστοιχίσεις που έχουν περιγραφεί με τη χρήση της D2RQ Map χρησιμοποιούνται από το εργαλείο Jena παράγοντας έναν RDF γράφο με όλη την πληροφορία που περιέχεται στη βάση δεδομένων. Με την D2RQ υπάρχει η δυνατότητα μεταχείρισης των σχεσιακών βάσεων δεδομένων σαν εικονικούς RDF γράφους στους οποίους μπορούν να υποβληθούν επερωτήσεις με τη χρήση της γλώσσας RDQL[2] [12]. Οι επερωτήσεις αυτές μεταφράζονται σε SQL επερωτήσεις, τα αποτελέσματα των οποίων μετατρέπονται και πάλι σε RDF τριπλέτες (Subject, Predicate, Object) και χρησιμοποιούνται από ανώτερα στρώματα του Jena Framework. Η D2RQ προσφέρει επίσης τη δυνατότητα δημοσίευσης του περιεχομένου μιας βάσης δεδομένων στον Σημασιολογικό Ιστό με τη χρήση του Joseki RDF εξυπηρετητή [13]. Όπως συμβαίνει με τα προαναφερθέντα εργαλεία και γλώσσες έτσι και η D2RQ αντιστοιχίζει τις στήλες των πινάκων της βάσης δεδομένων σε RDF properties. Χρησιμοποιεί όμως και επιπλέον στοιχεία τα οποία αναφέρονται ως “hint properties” που στόχο έχουν τη βελτίωση της απόδοσης των επερωτήσεων και την αύξηση της ταχύτητας εκτέλεσής τους. Ο ορισμός των αντιστοιχίσεων, όπως και στην D2R γλώσσα, γίνεται εξολοκλήρου από το χρήστη.



Εικόνα 4.8 Η αρχιτεκτονική που ακολουθείται από την D2RQ.

[1] Jena: ένα πλαίσιο Java για εφαρμογές Σημασιολογικού Ιστού.
[2] Η RDQL είναι γλώσσα επερωτήσεων σε RDF δεδομένα.

Στην εικόνα που ακολουθεί δίνεται ένα παράδειγμα χρήσης των “hint properties” στη γλώσσα D2RQ. Υποθέτουμε και πάλι ότι έχουμε έναν πίνακα στη βάση που περιέχει πληροφορίες για όλα τα βιβλία μιας βιβλιοθήκης : Book (isbn, title) και θέλουμε να εκφράσουμε με τη βοήθεια της γλώσσας D2RQ την αντιστοιχία της στήλης “title” αυτού του πίνακα με το datatype property “hasTitle” της οντολογίας. Όμως στην αντιστοίχιση αυτή μας ενδιαφέρουν τα βιβλία των οποίων οι τίτλοι δεν ξεπερνούν σε μήκος τους 15 χαρακτήρες. Γι’αυτό το λόγο χρησιμοποιείται το hint property “d2rq:valueMaxLength”.

```
... d2rq: DatatypePropertyBridge;
        d2rq:property  :hasTitle;
        d2rq:column  “Book.title”;
        …..
        d2rq:valueMaxLength “15”.
```

Εικόνα 4.9 Παράδειγμα χρήσης των “hint properties” στην D2RQ. Τα στοιχεία της γλώσσας εμφανίζονται με έντονα γράμματα.

4.5 Το σύστημα CUPID

Το Cupid [10, 14] είναι ένα σύστημα διαφορετικό από τα προηγούμενα ως προς την τεχνική την οποία ακολουθεί αλλά και ως προς τα σχήματα δεδομένων τα οποία προσπαθεί να ταιριάξει. Πιο συγκεκριμένα, το Cupid υλοποιεί αντιστοιχίσεις ανάμεσα σε σχεσιακά και σε XML μοντέλα δεδομένων, ενώ για την αναπαράσταση των μεταδεδομένων – τα οποία εξάγονται από τα προς αντιστοίχιση μοντέλα δεδομένων – χρησιμοποιεί το ER (Entity Relationship) μοντέλο.

Το Cupid είναι ένα σύστημα το οποίο συνδυάζει κάποιες από τις ήδη υπάρχουσες τεχνικές και ανακαλύπτει αντιστοιχίσεις ανάμεσα στα στοιχεία των προς αντιστοίχιση σχημάτων βασιζόμενο στα ονόματά τους, στους τύπους δεδομένων τους, στους περιορισμούς που υπάρχουν στα σχήματα αυτά καθώς και στη δομή τους.

Ο αλγόριθμος του Cupid χρησιμοποιεί μια δενδρική δομή προκειμένου να αναπαραστήσει τις διασυνδέσεις που υπάρχουν ανάμεσα στα στοιχεία ενός σχήματος (schema tree). Αφού αναπαραστήσει τα σχήματα με το συγκεκριμένο τρόπο, ο αλγόριθμος προχωρά στον υπολογισμό της ομοιότητας που υπάρχει ανάμεσα στα στοιχεία των σχημάτων. Ο υπολογισμός αυτός διακρίνεται σε δύο φάσεις. Στην πρώτη φάση υπολογίζεται ο βαθμός της *γλωσσικής ομοιότητας* (linguistic similarity – lsim) των στοιχείων των σχημάτων με βάση τα ονόματά τους, τους τύπους δεδομένων τους, τα

πεδία ορισμού τους, κτλ. Στη δεύτερη φάση υπολογίζεται ο βαθμός της *δομικής ομοιότητας* (structural similarity – ssim) των στοιχείων των σχημάτων. Ο υπολογισμός της ομοιότητας αυτής ανάμεσα σε δύο στοιχεία γίνεται σύμφωνα με την ομοιότητα των γειτονικών τους στοιχείων. Και οι δύο αυτοί βαθμοί ομοιότητας παίρνουν τιμές μέσα από το διάστημα [0,1]. Η σταθμισμένη ομοιότητα (weighted similarity - wsim) είναι μια συνάρτηση των δύο αυτών βαθμών (lsim και ssim) και υπολογίζεται με βάση τον ακόλουθο τύπο:

$$wsim = w_{struct} \times ssim + (1 - w_{struct}) \times lsim,$$

όπου το w_{struct} είναι μια σταθερά που παίρνει τιμές από το διάστημα [0,1]. Μια αντιστοίχιση δημιουργείται επιλέγοντας ένα ζευγάρι στοιχείων από τα σχήματα που έχει τη μεγαλύτερη σταθμισμένη ομοιότητα.

Η διαδικασία εύρεσης της γλωσσικής ομοιότητας περιλαμβάνει τα ακόλουθα βήματα:

- *Κανονικοποίηση*: Τα ονόματα των στοιχείων των σχημάτων υφίστανται ανάλυση (απόρριψη προθέσεων και άρθρων, ορισμός ακρωνύμιων, κτλ).
- *Κατηγοριοποίηση*: Τα στοιχεία των σχημάτων κατατάσσονται σε κατηγορίες ανάλογα με τους τύπους δεδομένων τους, την ιεραρχία του σχήματος και το γλωσσολογικό περιεχόμενο των ονομάτων τους.
- *Σύγκριση*: Στο βήμα αυτό συγκρίνονται τα μέρη στα οποία αναλύθηκαν τα ονόματα των στοιχείων στο πρώτο βήμα και γίνεται χρήση λεξικού συνωνύμων με στόχο τον υπολογισμό της γλωσσολογικής ομοιότητας μεταξύ δύο σχημάτων.

Το ταίριασμα των στοιχείων των σχημάτων σύμφωνα με το βαθμό της δομικής τους ομοιότητας γίνεται βάσει των παρακάτω κανόνων:

- Τα φύλλα των δέντρων είναι όμοια αν είναι όμοια γλωσσολογικά, αν έχουν τον ίδιο τύπο δεδομένων και αν τα αντίστοιχα γειτονικά τους στοιχεία είναι επίσης όμοια.
- Δύο εσωτερικά στοιχεία είναι όμοια αν είναι γλωσσολογικά όμοια και τα αντίστοιχα υποδέντρα με ρίζες τα στοιχεία αυτά είναι επίσης όμοια.
- Δύο εσωτερικά στοιχεία είναι δομικά όμοια αν τα φύλλα τους είναι όμοια.

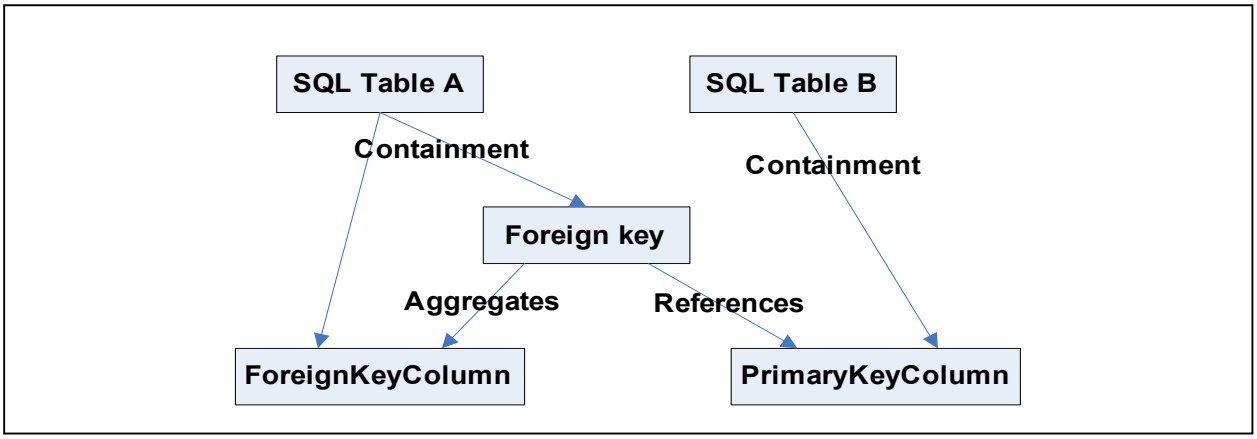
Η δομική ομοιότητα δύο φύλλων αρχικοποιείται στο βαθμό συμβατότητας των τύπων δεδομένων τους. Ορίζεται επίσης, ότι ένα φύλλο είναι ισχυρά συνδεδεμένο με ένα φύλλο του άλλου σχήματος αν η σταθμισμένη ομοιότητά τους ξεπερνά ένα κατώτατο όριο (threshold) το οποίο ορίζεται από τον χρήστη.

Τα σχήματα στα οποία εφαρμόζεται ο παραπάνω αλγόριθμος είναι δέντρα. Όμως, ο αλγόριθμος Cupid μπορεί να γενικευτεί ώστε να είναι δυνατή η εφαρμογή του και σε μοντέλα δεδομένων τα οποία δεν αναπαριστώνται με τη μορφή δέντρου. Ένα παράδειγμα τέτοιου μοντέλου είναι το σχεσιακό, το οποίο όπως γνωρίζουμε περιέχει αναφορικούς περιορισμούς (referential constraints), γεγονός που καθιστά δύσκολη την αναπαράστασή του σε δέντρο. Ο γενικευμένος αλγόριθμος Cupid διακρίνει τρεις τύπους διασύνδεσης των στοιχείων του σχήματος, οι οποίοι οδηγούν σε μία μη δενδρική μοντελοποίηση του σχήματος.

- *Περιεχόμενο (Containment)*: Δηλώνει ότι κάθε στοιχείο περιέχεται σ' ένα ακριβώς άλλο στοιχείο. Για παράδειγμα μια στήλη περιέχεται σε έναν πίνακα.
- *Σύνολο (Aggregation)*: Δηλώνει ένα σύνολο από στοιχεία. Για παράδειγμα το πρωτεύον κλειδί ενός πίνακα μπορεί να αποτελείται από ένα σύνολο από στήλες του πίνακα.
- *Προέρχεται από (IsDerivedFrom)*: Δηλώνει την πληροφορία που μπορεί να μοιράζονται δύο τύποι. Για παράδειγμα στο αντικειμενοστραφές μοντέλο δεδομένων η σχέση IsDerivedFrom συνδέει μια υποκλάση με την υπερκλάση της.

Όταν τα προς αντιστοίχιση σχήματα αναπαριστώνται σύμφωνα με το παραπάνω μοντέλο, η διαδικασία του γλωσσικού ταιριάσματος (linguistic matching) των στοιχείων των σχημάτων δεν επηρεάζεται. Αντιθέτως, η διαδικασία εύρεσης της δομικής ομοιότητας (structural matching) επηρεάζεται. Το γεγονός αυτό αντιμετωπίζεται με την προσθήκη στοιχείων (επομένως και κόμβων) που αναπαριστούν τις παραπάνω σχέσεις.

Ας υποθέσουμε για παράδειγμα (εικόνα 4.10) ότι σε μια βάση δεδομένων έχουμε δύο πίνακες (SQL Table A, SQL Table B) ανάμεσα στους οποίους υπάρχει ένας αναφορικός περιορισμός ακεραιότητας. Συγκεκριμένα ο SQL Table A περιέχει ένα ξένο κλειδί το οποίο αναφέρεται στο πρωτεύον κλειδί του SQL Table B. Για να αναπαρασταθεί η σχέση αυτή ανάμεσα στους δύο πίνακες έχει προστεθεί ένα επιπλέον στοιχείο-κόμβος, το "Foreign Key" στην εικόνα 4.10, το οποίο συνδέεται με το ξένο και το πρωτεύον κλειδί του SQL Table A και SQL Table B αντίστοιχα. Ο κόμβος αυτός αντιπροσωπεύει επίσης τη δυνατότητα συνένωσης (join) των πινάκων που λαμβάνουν μέρος στον περιορισμό αυτό.



Εικόνα 4.10 Τρόπος αναπαράστασης αναφορικών περιορισμών στα σχεσιακά σχήματα στο σύστημα Cupid

4.6 Σύνοψη

Στον πίνακα 4.1 συνοψίζονται τα βασικότερα χαρακτηριστικά των τεχνικών που περιγράφτηκαν στις προηγούμενες ενότητες.

Πίνακας 4.1 Χαρακτηριστικά των τεχνικών που ανακαλύπτουν ή εκφράζουν αντιστοιχίσεις ανάμεσα σε δύο σχήματα

	KAON Reverse	**R_2O**	**D2R**	**D2RQ**	**CUPID**
source schema	σχεσιακό	σχεσιακό	σχεσιακό	σχεσιακό	σχεσιακό, XML
target schema	οντολογία σε RDF	οντολογία σε RDF(S)/OWL	οντολογία σε OWL	οντολογία σε RDFS/OWL	σχεσιακό, XML
match cardinality	1:1 , 1:n , n:m	1:1 , 1:n , n:m	1:1 , 1:n , n:m	1:1 , 1:n , n:m	1:1 , 1:n
Μέτρηση ομοιότητας των στοιχείων των σχημάτων	Χρησιμοποιεί μεθόδους εύρεσης της λεξιλογικής (lexical) ομοιότητας ανάμεσα στα στοιχεία των σχημάτων.	Δεν υπολογίζει βαθμό ομοιότητας ανάμεσα στα στοιχεία. Ορίζει τις συνθήκες κάτω από τις οποίες είναι όμοια.	Δε χρησιμοποιεί μεθόδους μέτρησης της ομοιότητας μεταξύ των στοιχείων των δύο σχημάτων.	Δε χρησιμοποιεί μεθόδους μέτρησης της ομοιότητας μεταξύ των στοιχείων των δύο σχημάτων.	Χρησιμοποιεί μεθόδους εύρεσης της γλωσσικής και δομικής ομοιότητας ανάμεσα στα στοιχεία (linguistic – structural similarity).
Γλώσσα εφαρμογής	Οι κανόνες είναι ορισμένοι σε Java.	Χρησιμοποιεί ένα XML συντακτικό.	Χρησιμοποιεί ένα XML συντακτικό.	Χρησιμοποιεί ένα XML συντακτικό.	-
Παρέμβαση χρήστη	-Σε περιπτώσεις όπου μπορούν να εφαρμοστούν πολλοί κανόνες. -Όταν δεν	Ο ορισμός των αντιστοιχίσεων γίνεται από τον χρήστη.	Ο ορισμός των αντιστοιχίσεων γίνεται από τον χρήστη.	Ο ορισμός των αντιστοιχίσεων γίνεται από τον χρήστη.	Για τον ορισμό του κατώτερου ορίου βάσει του οποίου αποφασίζεται πότε ένα φύλλο του

	υπάρχουν μέθοδοι εξαγωγής των μετα-δεδομένων της βάσης.				δέντρου είναι ισχυρά συνδεδεμένο με ένα φύλλο του άλλου δέντρου.
Ειδικά χαρακτηριστικά	Είναι ολοκληρωμένο ημι-αυτόματο εργαλείο με γραφικό περιβάλλον.	-Είναι δυνατή η χρήση της από άλλα εργαλεία. -Επιτρέπει την εκτέλεση πράξεων της σχεσιακής άλγεβρας στους πίνακες πριν την αντιστοίχιση.	-Μπορεί να εκφράσει αντιστοιχίσεις ακόμα κι αν η πληροφορία κατανέμεται σε περισσότερες από μια βάσεις δεδομένων.	-Χρησιμοποιεί "hint properties" για να επιταχύνει τη διαδικασία εκτέλεσης των αντιστοιχίσεων. -Η χρήση της γίνεται μέσα στο Jena εργαλείο.	Μοντελοποιεί σε ιεραρχίες τα σχήματα.

<u>Σημείωση</u>: Ο όρος match cardinality δηλώνει το πλήθος των στοιχείων ενός συνόλου που συμμετέχει σε μια αντιστοίχιση. Πιο συγκεκριμένα, ένα στοιχείο ενός σχήματος μπορεί να συμμετέχει σε μηδέν, μία ή περισσότερες αντιστοιχίσεις (1:n) . Επίσης, σε μια αντιστοίχιση είναι δυνατόν ένα ή περισσότερα στοιχεία του ενός σχήματος να ταιριάζουν με ένα ή περισσότερα στοιχεία του άλλου σχήματος (1:1, m:n).

4.7 Προδιαγραφές ενός συστήματος ταιριάσματος δύο μοντέλων δεδομένων

Στον παρόν κεφαλαίο έγινε μια περιγραφή κάποιων συστημάτων και γλωσσών που στόχο έχουν την αντιστοίχιση ενός σχεσιακού σχήματος σε μια οντολογία. Κάθε ένα από αυτά τα συστήματα έχει πλεονεκτήματα και μειονεκτήματα. Για παράδειγμα, το KAON Reverse, σε αντίθεση με το Cupid, έχει φιλικό προς τον χρήστη περιβάλλον αλλά δεν χρησιμοποιεί ικανοποιητικούς αλγορίθμους εύρεσης της ομοιότητας ανάμεσα στα στοιχεία των προς αντιστοίχιση σχημάτων. Στις επόμενες παραγράφους συνοψίζονται οι προδιαγραφές που πρέπει να ικανοποιεί ένα σύστημα ώστε να είναι όσο το δυνατόν πλήρες και αποτελεσματικό.

Ένα από τα βασικά χαρακτηριστικά που πρέπει να έχει ένα σύστημα απεικόνισης ενός μοντέλου δεδομένων σε ένα άλλο, είναι το γραφικό περιβάλλον χρήσης (Graphical User

Interface, GUI). Ένα σύστημα με φιλικό προς το χρήστη περιβάλλον, όπως είναι το γραφικό, είναι πιο ελκυστικό και διευκολύνει σε μεγάλο βαθμό την κατανόηση της λειτουργίας για την οποία έχει σχεδιαστεί.

Είναι επίσης σημαντικό για ένα τέτοιο σύστημα να μην απαιτεί τη συνεχή παρέμβαση του χρήστη κατά τη διαδικασία της αντιστοίχισης των στοιχείων των δύο σχημάτων. Θα ήταν ιδανική η κατασκευή ενός εργαλείου που θα αντιστοίχιζε αυτόματα τα στοιχεία του ενός σχήματος στα στοιχεία του άλλου, όμως κάτι τέτοιο δεν είναι εφικτό αφού στα προς αντιστοίχιση σχήματα υπάρχουν πολύ σημαντικές διαφορές που πρέπει να ληφθούν υπόψη. Θα πρέπει λοιπόν το σύστημα να είναι ημι-αυτόματο και ο χρήστης να παρεμβαίνει για να δώσει λύσεις όταν προκύπτουν προβλήματα ασάφειας ή όταν δεν είναι δυνατή η αυτόματη εξαγωγή πληροφοριών που αφορούν στα προς αντιστοίχιση σχήματα (π.χ. δεν προσφέρουν όλοι οι JDBC drivers μεθόδους για την εξαγωγή των μετα-δεδομένων μιας βάσης δεδομένων).

Η διαδικασία αντιστοίχισης των στοιχείων της βάσης δεδομένων και της οντολογίας βασίζεται κυρίως στην ομοιότητα των στοιχείων αυτών, τόσο στην γλωσσική όσο και στη σημασιολογική, κι αυτό διότι τα προς αντιστοίχιση σχήματα μπορεί να έχουν κατασκευαστεί από διαφορετικά άτομα και άρα διαφορετικές λέξεις να περιγράφουν ακριβώς την ίδια οντότητα. Καθίσταται λοιπόν αναγκαία η χρήση ενός συνδυασμού από μεθόδους εύρεσης του βαθμού ομοιότητας ανάμεσα στα στοιχεία των σχημάτων.

Όσον αφορά τις οντολογίες που δεν αποτελούν απλές ταξινομίες αλλά περιέχουν σύνθετα αξιώματα και περιορισμούς είναι απαραίτητο το σύστημα να μπορεί να αναγνωρίζει τους περιορισμούς αυτούς ώστε να περιορίσει ανάλογα (μεταφράζοντας τους περιορισμούς σε SQL επερωτήσεις) τα δεδομένα της βάσης δεδομένων που θα μεταφερθούν στις αντίστοιχες κλάσεις της οντολογίας.

Τέλος, ένα σύστημα απεικόνισης ενός σχεσιακού σχήματος σε μια οντολογία πρέπει να μην εξαρτάται από το συστήμα στο οποίο έχει σχεδιαστεί η βάση δεδομένων αλλά και από το εργαλείο στο οποίο έχει υλοποιηθεί η οντολογία.

ΚΕΦΑΛΑΙΟ 5

ΜΕΘΟΔΟΛΟΓΙΑ ΑΠΕΙΚΟΝΙΣΗΣ ΣΧΕΣΙΑΚΟΥ ΜΟΝΤΕΛΟΥ ΣΕ ΟΝΤΟΛΟΓΙΑ

Η επιτυχία του Σημασιολογικού Ιστού εξαρτάται κυρίως από την εύκολη δημιουργία και χρήση σημασιολογικών δεδομένων. Το πρόβλημα όμως που αντιμετωπίζει ο Σημασιολογικός Ιστός είναι η ύπαρξη πολύ μικρής ποσότητας σημασιολογικών δεδομένων. Ως γνωστόν, οι οντολογίες αποτελούν δομικό στοιχείο του Σημασιολογικού Ιστού και χρησιμοποιούνται για την περιγραφή εννοιών και των σχέσεων που υπάρχουν ανάμεσά τους. Ο εμπλουτισμός των οντολογιών με πραγματικά δεδομένα θα συνέβαλλε ουσιαστικά στην αντιμετώπιση του προβλήματος που αντιμετωπίζει ο Σημασιολογικός Ιστός. Στηριζόμενοι στο γεγονός ότι οι βάσεις δεδομένων αποτελούν τον πιο ευρέως διαδεδομένο και χρησιμοποιημένο τρόπο αποθήκευσης, επεξεργασίας αλλά και ανάκτησης δεδομένων στον Παγκόσμιο Ιστό, ο εμπλουτισμός των οντολογιών με δεδομένα τα οποία είναι αποθηκευμένα σε βάσεις δεδομένων θα ήταν πολύ σημαντικός. Πριν όμως πραγματοποιηθεί αυτή η «μετακίνηση» των δεδομένων από τη σχεσιακή βάση στην οντολογία είναι απαραίτητο να αντιστοιχηθούν τα στοιχεία του πρώτου σχήματος σε εκείνα του δεύτερου.

Στο παρόν κεφάλαιο παρουσιάζεται η μεθοδολογία που ακολουθεί το RONTO προκειμένου να απεικονίσει αρχικά τα στοιχεία του σχεσιακού σχήματος σε εκείνα της οντολογίας ώστε να είναι εφικτή η «μετακίνηση» των δεδομένων από τη βάση στην οντολογία. Η μεθοδολογία αυτή δανείζεται κάποια στοιχεία από τα εργαλεία και τις γλώσσες που περιγράφτηκαν στο κεφάλαιο 4, προσπαθώντας παράλληλα να ικανοποιήσει τις απαιτούμενες προδιαγραφές που πρέπει να έχει ένα σύστημα αντιστοίχισης δύο σχημάτων. Η διαδικασία αντιστοίχισης των στοιχείων του σχεσιακού σχήματος στην οντολογία, που περιγράφεται στην ενότητα 5.1 εξαρτάται σε ένα μεγάλο βαθμό από την ομοιότητα των στοιχείων των δύο σχημάτων. Ο όρος ομοιότητα, αλλά και ο τρόπος μέτρησης της μελετάται στην ενότητα 5.3.

5.1 Αντιστοίχιση σχημάτων (Schema mapping)

Η διαδικασία απεικόνισης του σχεσιακού σχήματος σε οντολογία μπορεί να αναλυθεί στις ακόλουθες φάσεις:

- *Αντιστοίχιση κλάσεων (Concept Mapping):* Περιλαμβάνει την αντιστοίχιση των πινάκων της βάσης δεδομένων σε κλάσεις της οντολογίας.
- *Αντιστοίχιση των datatype-properties (Datatype-property mapping):* Περιλαμβάνει την αντιστοίχιση των γνωρισμάτων των πινάκων της βάσης δεδομένων σε datatype-properties της οντολογίας.
- *Αντιστοίχιση των object-properties (Object-property mapping):* Περιλαμβάνει την αντιστοίχιση των σχέσεων που υφίστανται ανάμεσα στους πίνακες της βάσης δεδομένων, όπως είναι οι περιορισμοί αναφορικής ακεραιότητας, σε object-properties της οντολογίας.

Στις ενότητες που ακολουθούν περιγράφονται αναλυτικά κάθε μια από τις παραπάνω φάσεις.

Πριν όμως προχωρήσουμε στην περιγραφή της απεικόνισης του σχεσιακού σχήματος στην οντολογία δίνουμε τους ορισμούς μερικών ενδιάμεσων στοιχείων που θα χρησιμοποιήσουμε στη συνέχεια.

Ορισμός 1 Μια υποψήφια κλάση (Candidate Concept, CC) για μια κλάση C της οντολογίας (CC_C) είναι μια υπάρχουσα ή εικονική σχέση (προερχόμενη από συνένωση ή προβολή σε υπάρχουσες σχέσεις) της βάσης δεδομένων η οποία είναι σημασιολογικά και λεξικογραφικά «όμοια» με την C και μπορεί να αντιστοιχηθεί σε αυτή. Για μια συγκεκριμένη κλάση είναι δυνατόν να υπάρχουν περισσότερες από μια υποψήφιες κλάσεις οι οποίες συνιστούν ένα σύνολο υποψηφίων κλάσεων (Candidate Concept Set, CCS) για την κλάση C (CCS_C).

$$CCS_C \equiv \bigcup_{i=1}^{n} \{CC_{Ci}\}$$

Ορισμός 2 Ένα υποψήφιο datatype-property (Candidate Datatype-property, CDP) για ένα datatype-property A (CDP_A) της οντολογίας είναι ένα γνώρισμα κάποιας σχέσης της βάσης δεδομένων, που δεν αποτελεί ξένο κλειδί, έχει όμοιο ή συμβατό τύπο δεδομένων με το A, είναι σημασιολογικά «όμοιο» με το A και είναι δυνατό να αντιστοιχηθεί σε αυτό. Αντίστοιχα με το σύνολο υποψηφίων κλάσεων ορίζεται και το σύνολο των υποψηφίων datatype-properties (Candidate Datatype-property Set, $CDPS_A$).

Ορισμός 3 Ένα υποψήφιο object-property (Candidate Object-property, COP) για ένα object-property P (COP_P) της οντολογίας είναι ένας περιορισμός αναφορικής ακεραιότητας του σχεσιακού σχήματος, που είναι λεξικογραφικά και σημασιολογικά «όμοιος» με το P και μπορεί να αντιστοιχηθεί σε αυτό. Η σχέση που περιέχει το ξένο κλειδί του περιορισμού θα πρέπει πρώτα να έχει αντιστοιχηθεί στο domain του P, ενώ η σχέση που περιέχει το πρωτεύον κλειδί που συμμετέχει στον περιορισμό θα πρέπει να έχει αντιστοιχηθεί στο range του P. Για αυτό το λόγο δεν μπορούμε να βρούμε τα υποψήφια object-properties μιας βάσης αν δεν έχουμε πρώτα ορίσει τις υποψήφιες κλάσεις αυτής. Ανάλογα με το σύνολο των υποψηφίων κλάσεων και το σύνολο των υποψηφίων datatype-properties, ορίζεται και το σύνολο των υποψηφίων object-properties (Candidate Object-property Set, $COPS_P$).

Κάθε στοιχείο των παραπάνω συνόλων «συνοδεύεται» και από μια τιμή, η οποία δηλώνει το βαθμό ομοιότητάς του με το αντίστοιχο στοιχείο της οντολογίας. Για να εισαχθεί ένα στοιχείο σε κάποιο από αυτά τα σύνολα θα πρέπει ο βαθμός ομοιότητάς του να ξεπερνά ένα κατώτατο όριο (threshold), το οποίο ορίζεται από το χρήστη.

5.1.1 Αντιστοίχιση Κλάσεων (Concept Mapping)

Στην ενότητα αυτή γίνεται μια εκτενής αναφορά στην διαδικασία με την οποία οι πίνακες της βάσης δεδομένων αντιστοιχίζονται στις κλάσεις της οντολογίας. Η διαδικασία αυτή δεν αφορά μόνο στην αντιστοίχιση καθενός ξεχωριστού πίνακα της βάσης δεδομένων σε μια κλάση της οντολογίας αλλά περιλαμβάνει και περιπτώσεις κατά τις οποίες ένας πίνακας προερχόμενος από τη συνένωση (join) δύο ή περισσότερων πινάκων της βάσης δεδομένων μπορεί να αντιστοιχιθεί επίσης σε μια κλάση της οντολογίας. Γι'αυτό η διαδικασία αντιστοίχισης πινάκων σε κλάσεις μπορεί να διαιρεθεί σε δύο επιμέρους κατηγορίες:

- Απλή αντιστοίχιση (direct mapping), στην οποία ένας πίνακας της βάσης αντιστοιχίζεται σε μια κλάση της οντολογίας.
- Σύνθετη αντιστοίχιση (complex mapping), όπου ένας πίνακας που αποτελεί συνένωση άλλων πινάκων αντιστοιχίζεται σε μια κλάση της οντολογίας.

Στο στάδιο αυτό είναι δυνατό κάποιοι πίνακες της βάσης να «αποκλειστούν» από τη διαδικασία της αντιστοίχισης. Αυτοί είναι οι πίνακες που εκφράζουν N:M σχέσεις ανάμεσα σε δύο άλλους διαφορετικούς πίνακες. Τυπικά, αυτό το είδος των πινάκων χαρακτηρίζεται από το γεγονός ότι περιέχει δύο γνωρίσματα τα οποία αποτελούν

πρωτεύον κλειδί της σχέσης και ξένα κλειδιά προς δύο άλλες σχέσεις. Αυτοί οι πίνακες δεν αναπαριστούν κάποια έννοια, γι'αυτό και δεν λαμβάνουν μέρος στην αντιστοίχιση κλάσεων. Ωστόσο συμμετέχουν στο object-property mapping καθώς αναπαριστούν σχέσεις ανάμεσα σε δύο έννοιες.

Στις ενότητες 5.1.1.1 και 5.1.1.2 περιγράφονται αντίστοιχα οι διαδικασίες της απλής και σύνθετης αντιστοίχισης.

5.1.1.1 Απλή αντιστοίχιση

Μετά την «απαλοιφή» των πινάκων που εκφράζουν N:M σχέσεις ανάμεσα σε άλλους πίνακες προχωρούμε στην απλή αντιστοίχιση. Στόχος του βήματος αυτού είναι η αντιστοίχιση των πινάκων της βάσης που αναπαριστούν κάποιες έννοιες στις αντίστοιχες κλάσεις της οντολογίας. Για να βρεθούν όλες οι πιθανές αντιστοιχίσεις των σχέσεων της βάσης σε κλάσεις της οντολογίας, ή αλλιώς όλες οι υποψήφιες κλάσεις (CC) για κάθε κλάση, χρησιμοποιούνται αλγόριθμοι εύρεσης της ομοιότητας (λεξικογραφικής και σημασιολογικής) που υπάρχει ανάμεσα στο όνομα μιας σχέσης και το όνομα μιας κλάσης. Η τελική απόφαση για το ποια από τις προτεινόμενες αντιστοιχίσεις θα ισχύσει λαμβάνεται από τον χρήστη.

Ο αλγόριθμος που εφαρμόζεται σε αυτό το βήμα περιγράφεται στην εικόνα 5.1

```
Tables = {all tables of the database} \ {tables which represent N:M relationships}
 Concepts = {all concepts of the ontology}
 CCSc ← ∅
 For each table t in Tables do
   For each concept c in Concepts do
      If (t is "similar" to c)
         then CCS_c ← CCS_c ∪ {t} that is, add t to the CCS of c
    End For
 End For
```

Εικόνα 5.1 Αλγόριθμος απλής αντιστοίχισης.

5.1.1.2 Σύνθετη Αντιστοίχιση

Κατά τη φάση απεικόνισης του σχεσιακού σχήματος στην οντολογία είναι πιθανή η χρήση σύνθετων συνενώσεων των σχέσεων (joins) ώστε να αντιστοιχηθούν σωστά σε κάποια υποψήφια κλάση της οντολογίας. Αυτό μπορεί να συμβεί στην περίπτωση που η

πληροφορία που φέρει μια κλάση κατανέμεται σε περισσότερους από έναν πίνακες της βάσης δεδομένων. Καθίσταται λοιπόν αναγκαία η γρήγορη εύρεση τόσο των πινάκων της βάσης δεδομένων όσο και των κλειδιών τους μέσω των οποίων μπορεί να συνδυαστεί μια συγκεκριμένη σχέση.

Στις ενότητες 5.1.1.2.1 και 5.1.1.2.2 περιγράφεται αντίστοιχα η μέθοδος εύρεσης όλων των δυνατών συνενώσεων μεταξύ των σχέσεων μιας βάσης δεδομένων και η διαδικασία εύρεσης όλων των σύνθετων αντιστοιχίσεων ανάμεσα στις σχέσεις αυτές και στις κλάσεις της οντολογίας.

5.1.1.2.1 Διαδικασία εύρεσης όλων των δυνατών συνενώσεων σε μια βάση δεδομένων

Γνωρίζουμε ότι η πράξη συνένωση χρησιμοποιείται για να συνδυαστούν σε εννιαίες πλειάδες κάποιες σχετιζόμενες πλειάδες από δύο σχέσεις. Μία πράξη συνένωσης πραγματοποιείται συγκρίνοντας δύο στήλες από δύο διαφορετικές σχέσεις ή από την ίδια σχέση που έχουν τον ίδιο τύπο δεδομένων. Στις περισσότερες περιπτώσεις, μια εκ των δύο αυτών στηλών είναι πρωτεύον κλειδί της σχέσης στην οποία ανήκει και η άλλη είναι ξένο κλειδί και αναφέρεται στην πρώτη, υπάρχει δηλαδή ένας αναφορικός περιορισμός ακεραιότητας (referential integrity constraint) μεταξύ των σχέσεων που συνενώνονται. Αυτό δεν αποτελεί κανόνα προκειμένου να συνενωθούν δύο πίνακες αλλά ένας τρόπος για να αποφευχθούν λανθασμένες εγγραφές στο αποτέλεσμα της συνένωσης.

Η μέθοδος λοιπόν που εφαρμόστηκε για την εύρεση των σχέσεων αλλά και των στηλών με τις οποίες μπορεί να συνδυαστεί μια δοθείσα σχέση βασίζεται στους περιορισμούς αναφορικής ακεραιότητας. Χρησιμοποιεί επίσης, τη θεωρία γράφων και τους αλγόριθμους που προσφέρει, πρώτα για την αναπαράσταση των περιορισμών αναφορικής ακεραιότητας και στη συνέχεια για την εύρεση όλων των δυνατών συνενώσεων που μπορούν να πραγματοποιηθούν ανάμεσα σε δύο ή περισσότερους πίνακες. Πιο συγκεκριμένα, ο αλγόριθμος περιλαμβάνει την κατασκευή ενός μη κατευθυνόμενου γράφου με τόσους κόμβους όσα είναι τα κλειδιά των πινάκων που συμμετέχουν στους περιορισμούς αναφορικής ακεραιότητας και τόσες ακμές όσες είναι το πλήθος αυτών των περιορισμών. Για παράδειγμα, αν ο πίνακας Α περιέχει ένα ξένο κλειδί x, το οποίο παίρνει τιμές από το πρωτεύον κλειδί y του πίνακα Β, τότε ο γράφος θα περιέχει δύο κόμβους που θα αντιστοιχούν στα ζεύγη «Α.x» και «Β.y» καθώς και μια ακμή που θα συνδέει τους δύο αυτούς κόμβους.

Για την υλοποίηση της μεθόδου χρησιμοποιήθηκε η JgraphT [2], μια βιβλιοθήκη της Java που παρέχει μαθηματικά αντικείμενα και αλγορίθμους της θεωρίας γράφων, καθώς και δύο ακόμα δομές για την αναπαράσταση των κόμβων και των ακμών, με ονόματα Vertex και GraphEdge αντίστοιχα (εικόνα 5.2). Ειδικότερα, κάθε κόμβος του γράφου αντιπροσωπεύεται από ένα αντικείμενο της δομής Vertex και περιγράφεται από το όνομα του ξένου ή πρωτεύοντος κλειδιού που συμμετέχει σε περιορισμό αναφορικής ακεραιότητας και από το όνομα του πίνακα στο οποίο βρίσκεται το συγκεκριμένο κλειδί. Ομοίως, κάθε ακμή του γράφου αναπαρίσταται από ένα αντικείμενο της δομής GraphEdge και χαρακτηρίζεται από τους δύο κόμβους που ενώνει.

```
class Vertex                    class GraphEdge
{                               {
   String tableName;               Vertex source;
   String keyName;                 Vertex target;
 }                               }
```

Εικόνα 5.2 Οι δομές Vertex και GraphEdge

Έχοντας αποφασίσει τον τρόπο περιγραφής των κόμβων και των ακμών του γράφου η μέθοδος προχωρά στην κατασκευή του γράφου με βάση το σχήμα (τα μεταδεδομένα) της σχεσιακής βάσης δεδομένων. Ο αλγόριθμος που εφαρμόζεται σε αυτό το βήμα περιγράφεται στην εικόνα 5.3.

Το επόμενο και τελευταίο βήμα μετά την αναπαράσταση των περιορισμών αναφορικής ακεραιότητας σε κόμβους και ακμές αφορά στην εύρεση όλων των δυνατών συνενώσεων μεταξύ των σχέσεων της βάσης. Αυτό μεταφράζεται στην εύρεση όλων των μονοπατιών από έναν δεδομένο κόμβο προς όλους τους υπόλοιπους. Αναλυτικότερα, δεδομένου ενός πίνακα Α της βάσης δεδομένων για να βρούμε τις σχέσεις με τις οποίες μπορεί να συνδυαστεί, πρέπει αρχικά να βρούμε τους κόμβους του γράφου για τους οποίους η τιμή της μεταβλητής tableName = Α. Εν συνεχεία, για κάθε έναν από αυτούς τους κόμβους η μέθοδος προχωρά στην εύρεση των συντομότερων μονοπατιών που τον συνδέουν με τους υπόλοιπους. Αυτό γίνεται με τη χρήση του αλγορίθμου Dijkstra [22] ο οποίος έχει υλοποιηθεί στην βιβλιοθήκη JGraphT. Η εικόνα 5.4 περιγράφει τον αλγόριθμο που υλοποιεί την προαναφερθείσα διαδικασία.

```
Vertices = ∅    /* το σύνολο των κόμβων του γράφου */
Edges = ∅       /* το σύνολο των ακμών του γράφου */
Tables = {all the tables of the database}
For each table t in Tables do
  For each foreign key f_k of table t do
    Vertex source = new Vertex (f_k.importedTable, f_k.importedColumn)
      /* Create a vertex source having as table name the name of the imported table of
         f_k(that is the name of t) and as column name the name of f_k */
    Vertex target = new Vertex (f_k.exportedTable, f_k.exportedColumn)
      /* create a vertex target having as table name the name of the exported table of
       f_k and as column name the primary key that is being referenced by f_k (that is
       the exported column of f_k */
    GraphEdge e = new GraphEdge (source, target)
      /*  Create a graphEdge e that links the vertices source and target,
          that is e = (source, target) */
    If source ∉ Vertices
      then  Vertices ← Vertices ∪ {source}
    If target ∉ Vertices
      then Vertices ← Vertices ∪ {target}
    If e ∉ Edges
      then Edges ← Edges ∪ {e}
  End for
End for
```

Εικόνα 5.3 Αλγόριθμος κατασκευής γράφου

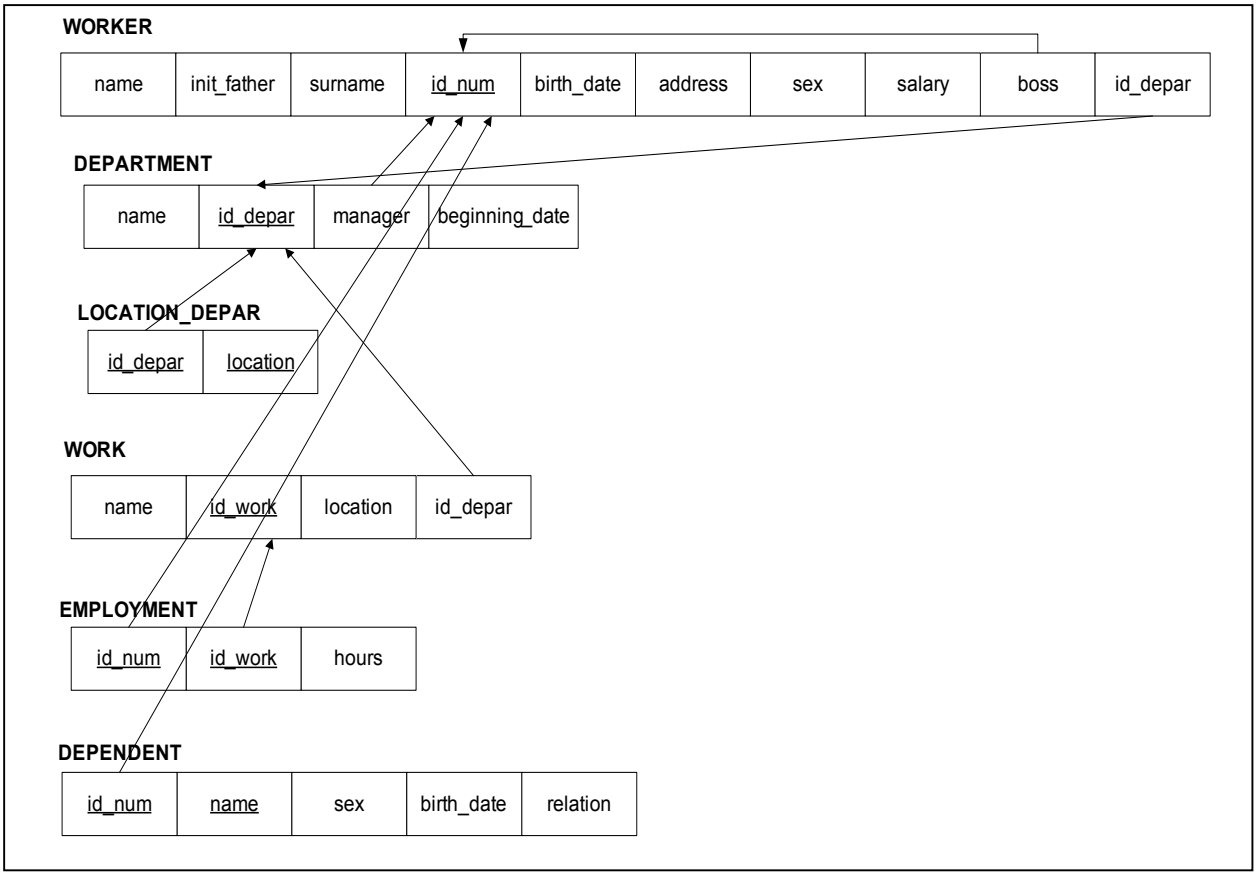
```
Vertices = {all the vertices of the graph}
Edges = {all the edges of the graph}
Tables = {all the tables of the database}
For each table t in Tables do
   Find all the vertices in Vertices that have the same table name with t.
   Add these vertices to a set U.
   For each vertex v in Vertices do
      For each vertex u in U do
        If (u ≠ v)
```

then Find the Dijkstra shortest path between *u* and *v*
End for
End for
End for

Εικόνα 5.4 Αλγόριθμος εύρεσης όλων των δυνατών συνενώσεων των σχέσεων μιας βάσης δεδομένων.

Εφαρμογή

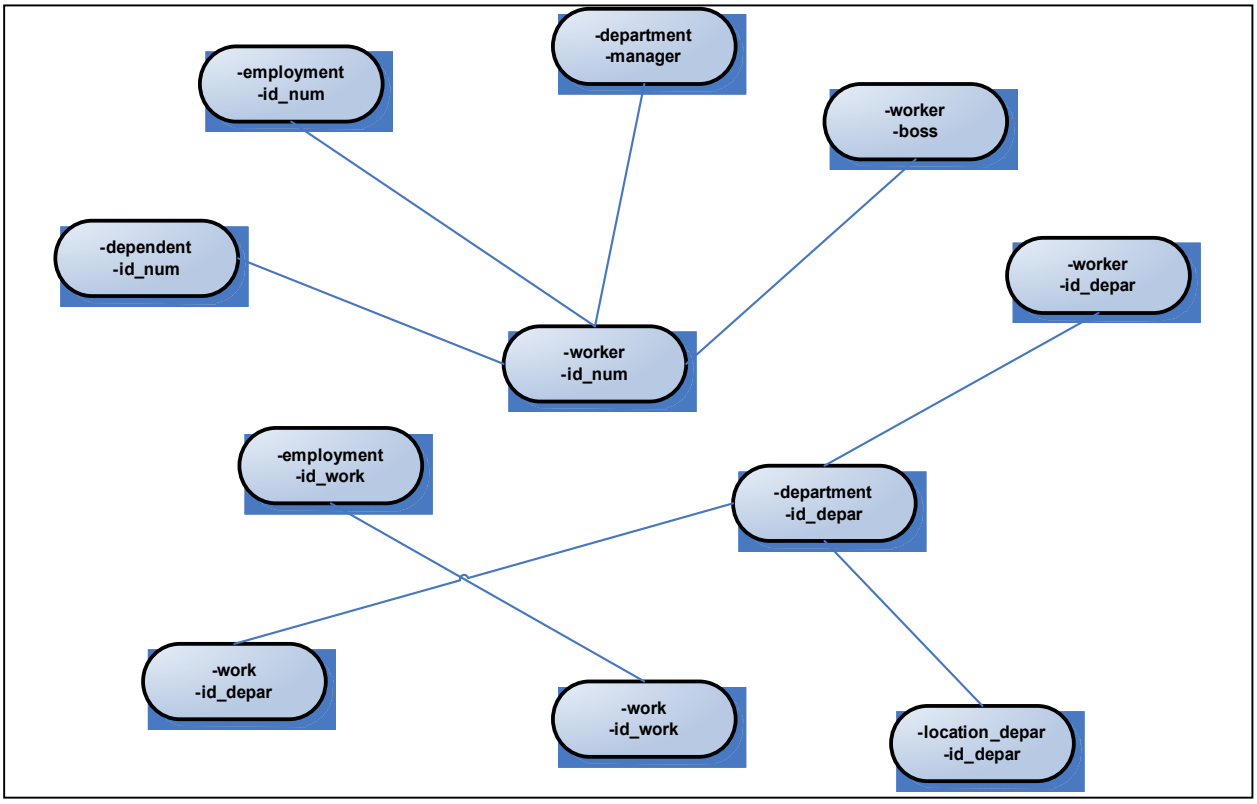
Στην εικόνα 5.5, δίνεται ένα σχεσιακό σχήμα βάσης δεδομένων με το όνομα COMPANY το οποίο περιλαμβάνει τους πίνακες WORKER, DEPARTMENT, LOCATION_DEPAR, WORK, EMPLOYMENT και DEPENDENT. Οι περιορισμοί αναφορικής ακεραιότητας παρουσιάζονται με τη χρήση ενός κατευθυνόμενου βέλους από το ξένο κλειδί προς το πρωτεύον κλειδί της σχέσης στην οποία αναφέρεται.



Εικόνα 5.5 Διάγραμμα σχήματος για το σχήμα της σχεσιακής βάσης δεδομένων COMPANY. Τα πρωτεύοντα κλειδιά είναι υπογραμμισμένα

Ο αριθμός των περιορισμών αναφορικής ακεραιότητας είναι 8 (όσα και τα κατευθυνόμενα βέλη), ενώ το πλήθος των κλειδιών που συμμετέχουν σ'αυτούς είναι 11 και είναι τα ακόλουθα: WORKER.id_num, WORKER.boss, WORKER.id_depar, DEPARTMENT.id_depar, DEPARTMENT.manager, LOCATION_DEPAR.id_depar, WORK.id_work, WORK.id_depar, EMPLOYMENT.id_num, EMPLOYMENT.id_work και DEPENDENT.id_num.

Ο αλγόριθμος της εικόνας 5.3 εφαρμόστηκε στο σχήμα της βάσης δεδομένων COMPANY και οδήγησε στη κατασκευή του ακόλουθου μη κατευθυνόμενου γράφου (εικόνα 5.6).



Εικόνα 5.6 Ο γράφος που κατασκευάστηκε με την εφαρμογή του αλγορίθμου της εικόνας 5.3 στη βάση δεδομένων COMPANY.

Κάθε κόμβος περιγράφεται από δύο αλφαριθμητικά, όνομα πίνακα και όνομα κλειδιού. Ο αριθμός των κόμβων του γράφου είναι 11 ενώ αποτελείται από 8 ακμές τόσες δηλαδή όσο και το πλήθος των αναφορικών περιορισμών ακεραιότητας.

Τέλος, η εφαρμογή του αλγορίθμου της εικόνας 5.4 θα δώσει για κάθε πίνακα του σχήματος όλες τις δυνατές συνενώσεις με τους άλλους πίνακες (εικόνα 5.7).

Table DEPARTMENT can be joined as following:
->department.manager - worker.id_num
->department.manager - worker.id_num - dependent.id_num
->department.manager - worker.id_num - employment.id_num
->location_depar.id_depar - department.id_depar
->work.id_depar - department.id_depar
->worker.id_depar - department.id_depar
->department.manager - worker.id_num - worker.boss

Table DEPENDENT can be joined as following:
->dependent.id_num - worker.id_num - department.manager
->dependent.id_num - worker.id_num
->dependent.id_num - worker.id_num - employment.id_num
->dependent.id_num - worker.id_num - worker.boss

Table EMPLOYMENT can be joined as following:
->employment.id_num - worker.id_num - department.manager
->employment.id_num - worker.id_num
->employment.id_num - worker.id_num - dependent.id_num
->employment.id_work - work.id_work
->employment.id_num - worker.id_num - worker.boss

Table LOCATION_DEPAR can be joined as following:
->location_depar.id_depar - department.id_depar
->location_depar.id_depar - department.id_depar - work.id_depar
->location_depar.id_depar - department.id_depar - worker.id_depar

Table WORK can be joined as following:
->employment.id_work - work.id_work
->work.id_depar - department.id_depar - location_depar.id_depar
->work.id_depar - department.id_depar
->work.id_depar - department.id_depar - worker.id_depar

```
Table WORKER can be joined as following:
->department.manager - worker.id_num
->worker.boss - worker.id_num - department.manager
->worker.boss - worker.id_num
->dependent.id_num - worker.id_num
->worker.boss - worker.id_num - dependent.id_num
->employment.id_num - worker.id_num
->worker.boss - worker.id_num - employment.id_num
->worker.id_depar - department.id_depar - location_depar.id_depar
->worker.id_depar - department.id_depar
->worker.id_depar - department.id_depar - work.id_depar
->worker.boss - worker.id_num
```

Εικόνα 5.7 Όλες οι δυνατές συνενώσεις των πινάκων της βάσης δεδομένων COMPANY

5.1.1.2.2 Διαδικασία σύνθετης αντιστοίχισης

Έχοντας λοιπόν βρει τους πίνακες με τους οποίους μπορεί να συνενωθεί κάθε πίνακας της βάσης δεδομένων προχωρούμε στη διαδικασία της σύνθετης αντιστοίχισης των πινάκων σε κλάσεις της οντολογίας.

Ο αλγόριθμος που εφαρμόζεται στο βήμα αυτό και περιγράφεται στην εικόνα 5.8, δημιουργεί για κάθε κλάση, που δεν αντιστοιχήθηκε από κάποιον πίνακα στην προηγούμενη φάση, υποψήφιες κλάσεις από συνενώσεις και προβολές μεταξύ σχέσεων της βάσης δεδομένων. Είναι προφανές πως στη φάση αυτή δεν συμμετέχουν οι σχέσεις της βάσης που δεν εμπλέκονται σε περιορισμούς αναφορικής ακεραιότητας, δηλαδή οι πίνακες που δεν περιέχουν κάποιο ξένο ή πρωτεύον κλειδί που να συμμετέχει σε έναν τέτοιο περιορισμό.

Ο αλγόριθμος της εικόνας 5.8 χωρίζεται σε δύο κυρίως βήματα. Στο $1^{ο}$ βήμα δομείται το σύνολο των υποψηφίων κλάσεων για κάθε κλάση της οντολογίας (CCS formation step, εικόνα 5.9), ενώ στο $2^{ο}$ βήμα «αφαιρούνται» όσες από τις υποψήφιες κλάσεις του $1^{ου}$ βήματος δεν μπορούν τελικά να αντιστοιχηθούν στην αντίστοιχη κλάση της οντολογίας (CCS refinement step, εικόνα 5.10). Στις ακόλουθες παραγράφους αναλύονται οι δύο αυτές φάσεις του αλγορίθμου.

Το $1^{ο}$ βήμα του αλγορίθμου βρίσκει για κάθε κλάση της οντολογίας, όλους τους πίνακες της βάσης δεδομένων που έχουν γνωρίσματα «όμοια» με τα datatype-properties της συγκεκριμένης κλάσης και τα τοποθετεί σε ένα σύνολο. Για παράδειγμα στην εικόνα 5.9 οι πίνακες R_1, R_2 και R_3 περιέχουν ένα γνώρισμα «όμοιο» με το DP_2 της κλάσης C_1 και τοποθετούνται στο σύνολο T^2_1. Το ίδιο συμβαίνει και με τα υπόλοιπα datatype-properties της κλάσης C_1. Έχοντας δημιουργήσει το αντίστοιχο σύνολο για κάθε datatype-property της κλάσης, ο αλγόριθμος προχωρά στην δημιουργία του συνόλου των υποψηφίων κλάσεων συνενώνοντας διαφορετικούς πίνακες από διαφορετικά σύνολα με βάση τους περιορισμούς αναφορικής ακεραιότητας που υπάρχουν ανάμεσά τους. Σύμφωνα με την εικόνα 5.9 γίνονται δύο συνενώσεις πινάκων, ανάμεσα στις σχέσεις R_1 και R_3 από τα σύνολα T^1_1 και T^2_1 αντίστοιχα, και μεταξύ των σχέσεων R_4, R_5 και R_2 από τα σύνολα T^1_1, T^3_1 και T^2_1 αντίστοιχα.

Στο $2^{ο}$ βήμα του αλγορίθμου παίρνουν μέρος τα object-properties κάθε κλάσης της οντολογίας. Συγκεκριμένα, για κάθε object-property OP_j μιας κλάσης C_i αφαιρούνται από το σύνολο των υποψηφίων κλάσεων του domain του OP_j (δηλαδή από το CCS_{Ci}) όλες οι υποψήφιες κλάσεις που δεν περιέχουν ξένο κλειδί που να αναφέρεται στο πρωτεύον κλειδί κάποιας υποψήφιας κλάσης του συνόλου των υποψηφίων κλάσεων του range του OP_j. Αντίστοιχα από το σύνολο των υποψηφίων κλάσεων του range του OP_j αφαιρούνται οι υποψήφιες κλάσεις που δεν περιέχουν πρωτεύον κλειδί στο οποίο να αναφέρεται ένα ξένο κλειδί από το CCS_{Ci}. Στην εικόνα 5.10 έχει αφαιρεθεί από το CCS_D το CC^3_D και από το CCS_R το CC^3_R.

*/*CCS formation step*/*

Concepts = {all concepts of the ontology} \ {concepts mapped in previous step}

Tables = {all tables of the database} \ {tables not participating in referential constraints}

For each concept C_i in Concepts **do**

For each datatype-property DP_j of C_i **do**

Find all attributes of tables in Tables which are similar to DP_j

Add the corresponding tables in a set T^j_i along with the degree of similarity

End For

CCS_{Ci} = {all possible natural projection-joins between different relations from different T^j_i}

Assign higher degree of similarity to the elements of CCS_{Ci} with less participating tables

End For

*/*CCS refinement step*/*

For each concept C_i in Concepts **do**

For each object-property OP_j of C_i **do**

D = Domain (OP_j) /*D = C_i*/

R = Range (OP_j)

Do

CCS_D = CCS_D \ {CCs not containing foreign key in referential constraint with elements of CCS_R}

CCS_R = CCS_R \ {CCs not containing primary keys in referential constraint with elements of CCS_D}

Until (all $CC_D \in CCS_D$ contain foreign key in referential constraint with an element of CCS_R)

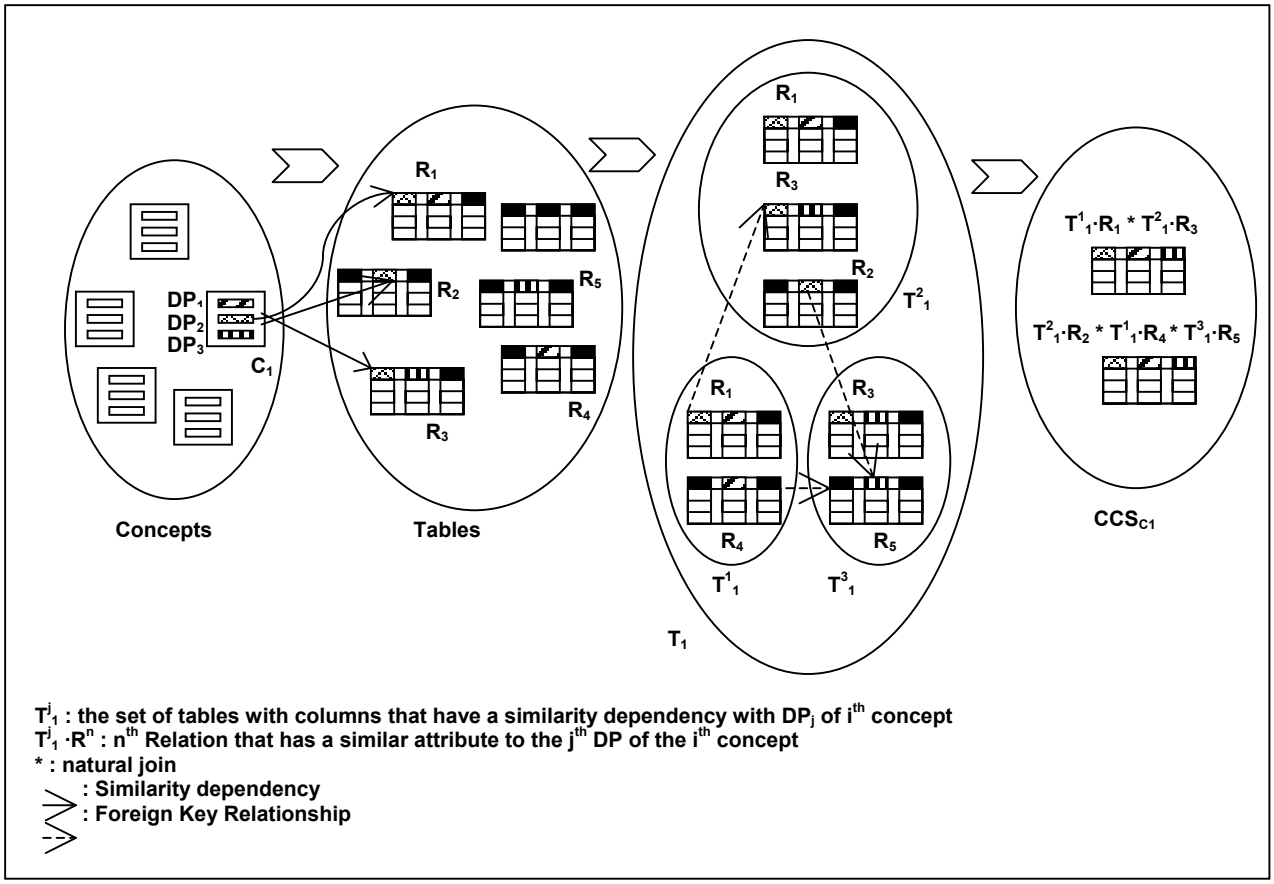
AND

(all $CC_R \in CCS_D$ contain primary key in referential constraint with an element of CCS_D)

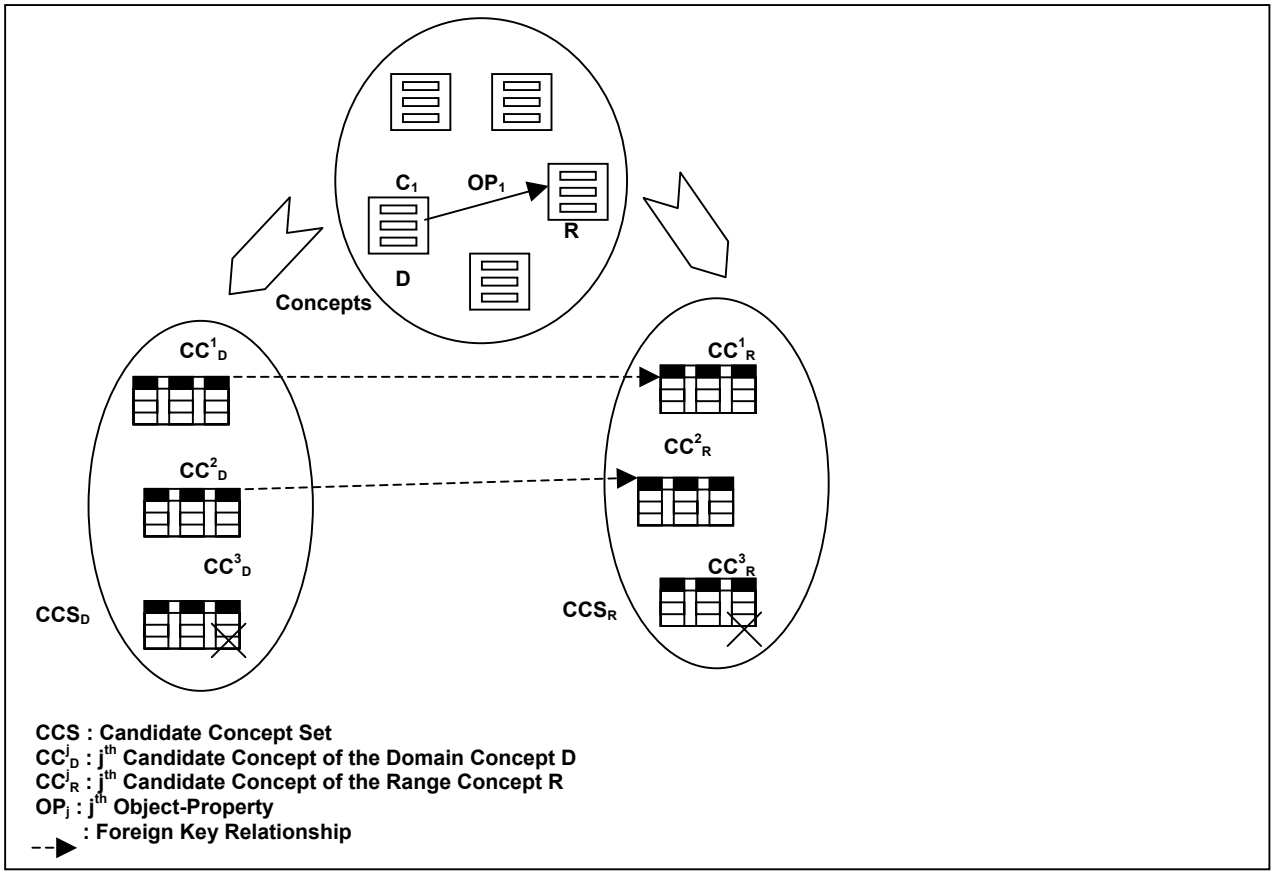
End For

End For

Εικόνα 5.8 Αλγόριθμος σύνθετης αντιστοίχισης



Εικόνα 5.9 Παράδειγμα εφαρμογής του αλγορίθμου σύνθετης αντιστοίχισης (1ο βήμα)



Εικόνα 5.10 Παράδειγμα εφαρμογής του αλγορίθμου σύνθετης αντιστοίχισης (2ο βήμα)

5.1.2 Αντιστοίχιση datatype-properties (Datatype-property mapping)

Ένα από τα βήματα της απεικόνισης του σχεσιακού μοντέλου σε οντολογία αφορά στην αντιστοίχιση των γνωρισμάτων των σχέσεων της βάσης δεδομένων στα datatype-properties της οντολογίας (datatype property mapping). Προκειμένου να επιτευχθεί μια τέτοια αντιστοίχιση θα πρέπει να πληρούνται κάποιες προϋποθέσεις. Μία από αυτές είναι και η ακόλουθη: ο xml schema τύπος δεδομένων (XML Schema datatype-xsd) στον οποίο έχει αντιστοιχηθεί ο sql τύπος δεδομένων της στήλης πρέπει να ελέγχει αν είναι όμοιος ή «περίπου» όμοιος με τον xsd που έχει οριστεί σαν range του υποψήφιου προς αντιστοίχηση datatype-property. Το ερώτημα που τίθεται σ'αυτό το σημείο είναι ποιοι xsd τύποι δεδομένων ταιριάζουν μεταξύ τους και ποιος είναι ο βαθμός ομοιότητάς τους;

Ας θεωρήσουμε για παράδειγμα τον πίνακα Worker της βάσης δεδομένων COMPANY. Ο πίνακας αυτός περιέχει το γνώρισμα *salary* στο οποίο αποθηκεύονται οι μισθοί των εργαζομένων της εταιρείας σε ακέραια μορφή (εικόνα 5.11). Θεωρούμε επίσης ότι η προς αντιστοίχιση οντολογία περιλαμβάνει ένα datatype-property με όνομα *hasSalary*, το range του οποίου είναι xsd:float (εικόνα 5.12).

```
create table worker (
        name varchar,
        surname varchar,
        ……..,
        salary int,
        …… );
```

Εικόνα 5.11 Ο πίνακας Worker

```
<owl:DatatypeProperty rdf:ID="hasSalary">
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
   <rdfs:domain rdf:resource="#Worker"/>
 </owl:DatatypeProperty>
```

Εικόνα 5.12 Ο ορισμός του datatype-property hasSalary.

Με βάση τους παραπάνω ορισμούς, θα μπορούσε να αντιστοιχηθεί το γνώρισμα *salary* στο datatype-property *hasSalary* ; Όπως παρατηρούμε το μόνο που θα μπορούσε να

αποτελέσει «εμπόδιο» σε μια τέτοια αντιστοίχιση είναι το γεγονός ότι αναπαρίστανται από διαφορετικούς τύπους δεδομένων (int και float). Γνωρίζουμε όμως ότι το σύνολο των δεκαδικών αριθμών είναι υπερσύνολο των ακεραίων, αφού ένας ακέραιος μπορεί να μετατραπεί σε δεκαδικό αν προστεθεί σ'αυτόν η υποδιαστολή και τουλάχιστον ένα μηδενικό. Με άλλα λόγια, θα μπορούσαμε να πούμε ότι ο float είναι ένας τύπος δεδομένων συμβατός με τον int, οπότε είναι δυνατή η αντιστοίχιση του γνωρίσματος *salary* στο datatype-property *hasSalary*.

Εκτός όμως από τη συμβατότητα των XML Schema τύπων δεδομένων σημαντικό ρόλο στην απεικόνιση των γνωρισμάτων των σχέσεων της βάσης σε datatype-properties της οντολογίας παίζει η ομοιότητα που υπάρχει ανάμεσα στα ονόματα των γνωρισμάτων και στα ονόματα των datatype-properties καθώς και οι αντιστοιχίσεις των πινάκων σε κλάσεις που προέκυψαν από την αντιστοίχιση κλάσεων. Όσον αφορά το προηγούμενο παράδειγμα προκειμένου να επιτευχθεί μια αντιστοίχιση ανάμεσα στο γνώρισμα *salary* και στο datatype-property *hasSalary*, θα πρέπει να ελεγχθεί η «ομοιότητα» των ονομάτων τους καθώς και το αν ο πίνακας *worker* έχει αντιστοιχηθεί στην κλάση *Worker* που αποτελεί και το domain του datatype-property *hasSalary*.

Πριν προχωρήσουμε στην παρουσίαση του τρόπου με τον οποίο τα γνωρίσματα των πινάκων της βάσης αντιστοιχίζονται σε datatype-properties της οντολογίας πρέπει να σημειώσουμε ότι από τη διαδικασία αυτή εξαιρούνται εκείνα τα γνωρίσματα τα οποία χρησιμοποιούνται ως πρωτεύοντα κλειδιά αυτόματης αρίθμησης (auto increment). Τα πρωτεύοντα κλειδιά αυτόματης αρίθμησης καταχωρούν αυτόματα έναν αύξοντα αριθμό κάθε φορά που προστίθεται μια εγγραφή στον πίνακα και ως εκ τούτου δεν φέρουν καμιά σημασιολογία. Στην παρούσα διαδικασία δεν συμμετέχουν επίσης και τα ξένα κλειδιά των σχέσεων καθώς αυτά παίρνουν μέρος στην αντιστοίχιση των object-properties (ενότητα 5.1.3).

Στην ενότητα 5.1.2.1 που ακολουθεί παρουσιάζεται η συμβατότητα που υφίσταται στους αριθμητικούς τύπους δεδομένων καθώς και ένας τρόπος εύρεσης του πόσο συμβατός είναι ένας τύπος δεδομένων με κάποιον άλλον, δηλαδή του βαθμού συμβατότητας δύο τύπων δεδομένων, ενώ στην ενότητα 5.1.2.2 περιγράφεται ο αλγόριθμος απεικόνισης των γνωρισμάτων των σχέσεων της βάσης σε datatype-properties της οντολογίας. Η διαδικασία που περιγράφεται στην ενότητα 5.1.2.1 μπορεί να εφαρμοστεί και σε μη αριθμητικούς τύπους δεδομένων.

5.1.2.1 Συμβατοί XML Schema τύποι δεδομένων

Στον πίνακα 5.1 δίνονται όλοι οι αριθμητικοί xml schema τύποι δεδομένων καθώς και τα πεδία ορισμού τους [3].

Πίνακας 5.1 Αριθμητικοί xsd τύποι δεδομένων

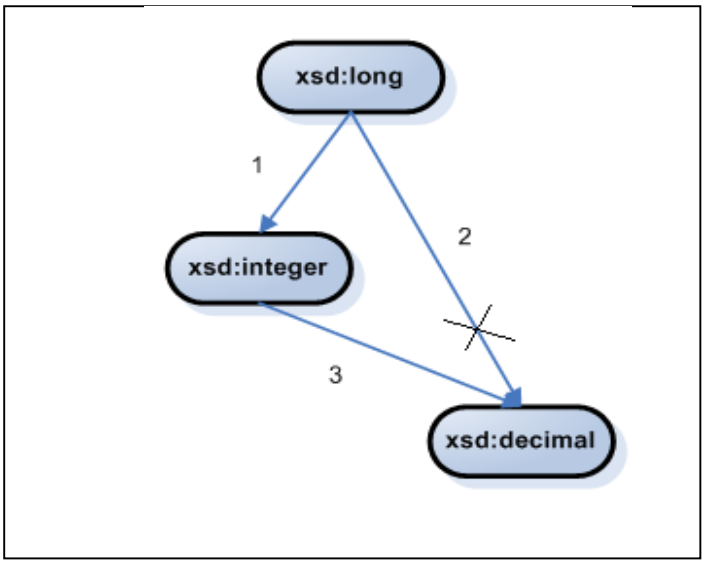
XSD	ΠΕΡΙΓΡΑΦΗ
byte	127 to-128. Sign is omitted, “+” assumed. Example: -1, 0, 126, +100
Decimal	Arbitrary precision decimal numbers. Sign omitted, “+” is assumed. Leading and trailing zeroes are optional. If the fractional part is zero, the period and following zero(es) can be omitted.
Double	Double-precision 64-bit floating point type - legal literals {0, -0, INF, -INF and NaN} Example, -1E4, 12.78e-2, 12 and INF
Float	32-bit floating point type - legal literals {0, -0, INF, -INF and NaN} Example, -1E4, 1267.43233E12, 12.78e-2, 12 and INF
Int	2147483647 to -2147483648. Sign omitted, “+” is assumed. Example: -1, 0, 126789675, +100000. §3.3.
Integer	Integer or whole numbers - Sign omitted, “+” is assumed. Example: -1, 0, 12678967543233, +100000.
Long	9223372036854775807 to - 9223372036854775808. Sign omitted, “+” assumed. Example: -1, 0, 12678967543233, +100000.
negativeInteger	Infinite set {...,-2,-1}. Example: -1, -12678967543233, -100000.
nonNegativeInteger	Infinite set {0, 1, 2,...}. Sign omitted, “+” assumed. Example: 1, 0, 12678967543233, +100000.
nonPositiveInteger	Infinite set {...,-2,-1,0}. Example: -1, 0, - 126733, -100000.
positiveInteger	Infinite set {1, 2,...}. Optional “+” sign,. Example: 1, 12678967543233, +100000.
Short	32767 to -32768. Sign omitted, “+” assumed. Example: -1, 0, 12678, +10000.
unsignedByte	0 to 255. a finite-length Example: 0, 126, 100.
unsignedInt	0 to 4294967295 Example: 0, 1267896754, 100000
unsignedLong	0 to 18446744073709551615. Example: 0, 12678967543233,
unsignedShort	0 to 65535. Example: 0, 12678, 10000.

Με τη βοήθεια του παραπάνω πίνακα κατασκευάστηκε ο πίνακας 5.2 ο οποίος δείχνει σε ποιους τύπους μπορεί να «μετατραπεί» -με ποιους τύπους είναι συμβατός- καθένας από τους τύπους δεδομένων του πίνακα 5.1.

Σύμφωνα με τον πίνακα 5.2 το σύνολο των τύπων που είναι συμβατοί, για παράδειγμα, με τον xsd:positiveInteger είναι το {xsd:nonNegativeInteger, xsd:integer, xsd:decimal}.

Έχοντας βρει τους xsd τύπους δεδομένων με τους οποίους είναι συμβατός κάποιος xsd προχωρούμε στην εύρεση του βαθμού συμβατότητας δύο xsd τύπων δεδομένων. Η τακτική που ακολουθείται για τη λύση του προβλήματος αυτού περιλαμβάνει την απεικόνιση της πληροφορίας που φέρει ο πίνακας 5.2 σε ένα δέντρο και στην συνέχεια την εφαρμογή κάποιας μετρικής προκειμένου να αποφασιστεί ο βαθμός ομοιότητας δύο xml schema τύπων δεδομένων.

Το δέντρο της εικόνας 5.14 δείχνει τις συμβατότητες που υπάρχουν ανάμεσα στους xsd τύπους δεδομένων χρησιμοποιώντας όσο το δυνατόν λιγότερες ακμές. Για παράδειγμα, στο σχήμα 5.13 φαίνεται ο τρόπος απεικόνισης των συμβατοτήτων για τους τύπους xsd:long και xsd:integer. Σύμφωνα με το σχήμα αυτό οι ακμές 1 και 2 δηλώνουν ότι οι xsd:integer και xsd:decimal αντίστοιχα είναι συμβατοί με τον xsd:long ενώ η ακμή 3 δηλώνει ότι ο xsd:decimal είναι επίσης συμβατός με τον xsd:integer. Οι συμβατότητες αυτές μπορούν να αναπαρασταθούν και χωρίς την ακμή 2, γι'αυτό και αφαιρείται. Το δέντρο που προκύπτει με την αφαίρεση αυτή δηλώνει ότι ο xsd:decimal είναι συμβατός με τον xsd:integer και τον xsd:long καθώς επίσης και το γεγονός ότι ο xsd:integer είναι συμβατός με τον xsd:long. Η συμβατότητα των XML Schema τύπων δεδομένων είναι κατευθυντική και όχι συμμετρική. Αυτό σημαίνει ότι ενώ ο xsd:decimal είναι συμβατός με τον xsd:long, αφού ένας ακέραιος μεγάλου μήκους αριθμός μπορεί να μετατραπεί σε δεκαδικό αν προστεθεί σ'αυτόν η υποδιαστολή και τουλάχιστον ένα μηδενικό, το αντίστροφο δεν ισχύει. Για αυτό το λόγο στο γράφο της εικόνας 5.13 χρησιμοποιούνται κατευνόμενες ακμές.

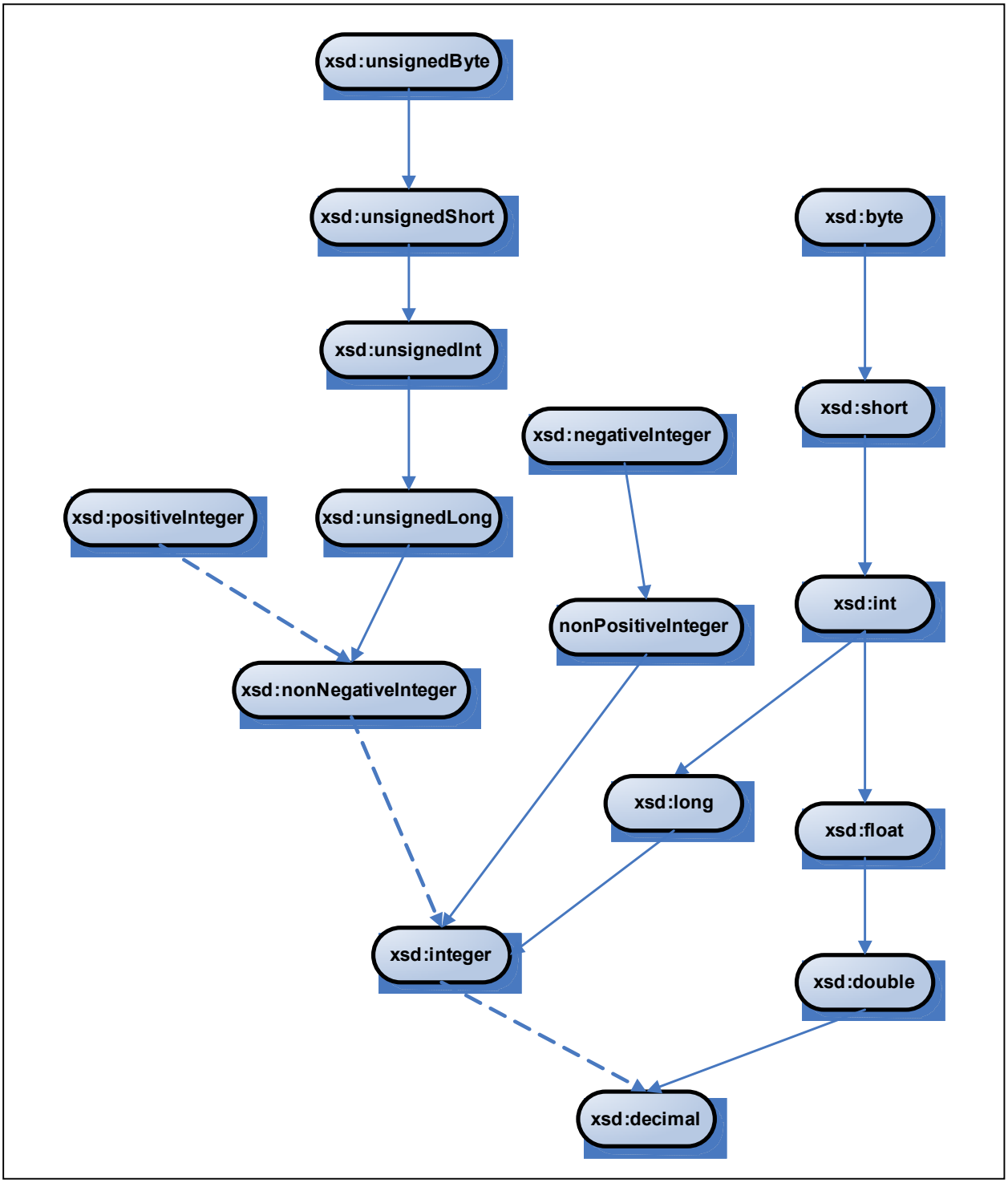


Εικόνα 5.13 Διαδικασία κατασκευής του δέντρου συμβατοτήτων

Πίνακας 5.2 Συμβατοί xml schema τύποι δεδομένων για κάθε έναν xsd της 1ης στήλης.

xsd	Compatible xml schema datatypes				
xsd:byte	xsd:int	xsd:long	xsd:short	xsd:float	xsd:double
	xsd:integer	xsd:decimal			
xsd:decimal					
xsd:double	xsd:decimal				
xsd:float	xsd:double	xsd:decimal			
xsd:int	xsd:long	xsd:float	xsd:double	xsd:integer	xsd:decimal
xsd:integer	xsd:decimal				
xsd:long	xsd:integer	xsd:decimal			
xsd:negativeInteger	xsd:nonPositiveInteger	xsd:integer	xsd:decimal		
xsd:nonNegativeInteger	xsd:integer	xsd:decimal			
xsd:nonPositiveInteger	xsd:integer	xsd:decimal			
xsd:positiveInteger	xsd:nonNegativeInteger	xsd:integer	xsd:decimal		
xsd:short	xsd:int	xsd:long	xsd:float	xsd:double	xsd:integer
	xsd:decimal				
xsd:unsignedByte	xsd:unsignedShort	xsd:unsignedInt	xsd:unsignedLong	xsd:integer	xsd:nonNegativeInteger
	xsd:decimal				
xsd:unsignedInt	xsd:integer	xsd:unsignedLong	xsd:nonNegativeInteger		
xsd:unsignedLong	xsd:nonNegativeInteger	xsd:integer	xsd:decimal		
xsd:unsignedShort	xsd:unsignedInt	xsd:integer	xsd:unsignedLong	xsd:decimal	xsd:nonNegativeInteger

Το δέντρο που κατασκευάζεται με τη βοήθεια του πίνακα 2 περιλαμβάνει τόσους κόμβους όσοι είναι και οι xsd τύποι δεδομένων του πίνακα 1 (εικόνα 5.14).



Εικόνα 5.14 Γραφική αναπαράσταση της πληροφορίας του πίνακα 5.2

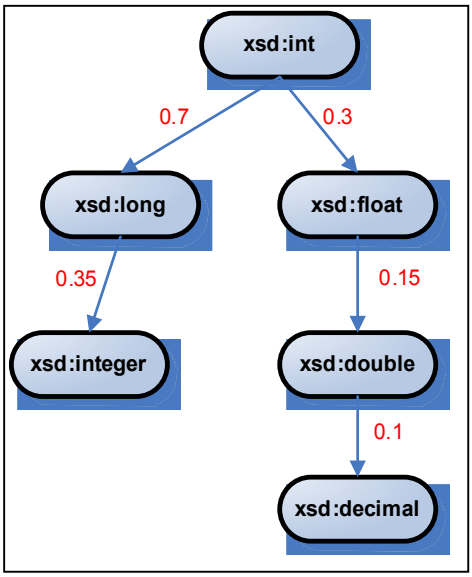
Για να βρούμε το σύνολο των τύπων με τους οποίους είναι συμβατός ένας συγκεκριμένος xsd, ακολουθούμε το μονοπάτι με αφετηρία τον αντίστοιχο κόμβο του δέντρου, προς όλους τους υπόλοιπους κόμβους. Για παράδειγμα, αν ακολουθήσουμε το διακεκομμένο μονοπάτι, στην εικόνα 5.14, που έχει ως αφετηρία τον

xsd:positiveInteger παίρνουμε όλους τους συμβατούς με αυτόν xml schema τύπους δεδομένων, δηλαδή τους xsd:nonNegativeInteger, xsd:integer και xsd:decimal.

Ο βαθμός συμβατότητας δύο xml schema τύπων δεδομένων είναι ένας αριθμός μεταξύ του διαστήματος (0,1]. Ο μέγιστος βαθμός συμβατότητας μεταξύ δύο διαφορετικών xml schema τύπων δεδομένων είναι 1. Όσο η τιμή του συγκεκριμένου βαθμού πλησιάζει το 1, τόσο πιο συμβατοί μπορούν να θεωρηθούν οι αντίστοιχοι xml schema τύποι δεδομένων. Το δέντρο της εικόνας 5.14, εκτός από την πληροφορία που φέρει για την ύπαρξη ή μη της συμβατότητας μεταξύ των xsd τύπων δεδομένων, δίνει και μια πρώτη διαισθητική εικόνα για τους βαθμούς συμβατότητας. Ας πάρουμε για παράδειγμα τις διακεκομμένες ακμές της εικόνας 5.14. Ο xsd:nonNegativeInteger έχει μεγαλύτερο βαθμό συμβατότητας με τον xsd:positiveInteger από τον xsd:integer και αυτό διότι το μονοπάτι από τον xsd:positiveInteger στον xsd:nonNegativeInteger είναι μικρότερου μήκους από το μονοπάτι με αφετηρία τον xsd:nonNegativeInteger και προορισμό τον xsd:integer.

Ας δούμε στη συνέχεια με τη βοήθεια δύο παραδειγμάτων τον τρόπο απόδοσης τιμών στους βαθμούς συμβατότητας των xsd τύπων δεδομένων.

Παράδειγμα 1 Θεωρούμε το υποδέντρο της εικόνας 5.15 που έχει ρίζα τον κόμβο xsd:int (εικόνα 5.15). Ακολουθεί η διαδικασία απόδοσης τιμών στους βαθμούς συμβατότητας μεταξύ της ρίζας του δέντρου και των κόμβων-παιδιών του.



Εικόνα 5.15 Υποδέντρο με ρίζα τον xsd:int

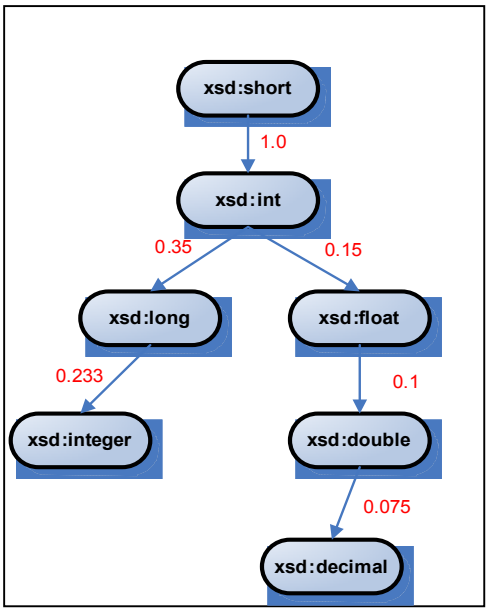
Ξεκινάμε από το πρώτο επίπεδο στο οποίο βρίσκονται οι κόμβοι xsd:long και xsd:float. Θεωρούμε ότι ο xsd:int είναι περισσότερο όμοιος με τον xsd:long από ότι με τον xsd:float, αφού οι xsd:int και xsd:long αναπαριστούν ακέραιους αριθμούς διαφορετικού μήκους ενώ ο xsd:float αναπαριστά δεκαδικούς αριθμους, και δίνουμε την τιμή 0.7 στον αντίστοιχο βαθμό συμβατότητας. Συνεπώς ο βαθμός συμβατότητας του xsd:float με τον xsd:int θα είναι: 1-0.7=0.3.

Προχωρούμε στο δεύτερο επίπεδο του δέντρου και στους κόμβους xsd:integer και xsd:double. Ο βαθμός συμβατότητας του xsd:integer με τον xsd:int

εξαρτάται από τον βαθμό συμβατότητας του xsd:long με τον xsd:int. Η μετρική που συμβατότητας του xsd:integer με τον xsd:int είναι: βαθμός συμβατότητας$_{xsd:integer}$ = [h/(h+1)] · βαθμός συμβατότητας$_{xsd:long}$, όπου h το βάθος του κόμβου xsd:long. Έχουμε λοιπόν: βαθμός συμβατότητας$_{xsd:integer}$ = (1/1+1)·0.7=0.35

Εφαρμόζοντας την ίδια μετρική για τους κόμβους xsd:double και xsd:decimal παίρνουμε τις τιμές που αναγράφονται στις αντίστοιχες ακμές του δέντρου της εικόνας 5.15. Όσον αφορά τον xsd:decimal πρέπει να σημειωθεί ότι ο υπολογισμός του βαθμού συμβατότητάς του με τον xsd:int βασίστηκε στο βαθμό συμβατότητας του xsd:double και όχι του xsd:integer. Η επιλογή αυτή στηρίζεται στο γεγονός ότι οι τύποι xsd:double και xsd:decimal αναπαριστούν ίδιου τύπου αριθμούς, δηλαδή δεκαδικούς, ενώ ο xsd:integer αναπαριστά ακέραιους.

Παράδειγμα 2 Θεωρούμε το υποδέντρο της εικόνας 5.16 που έχει ρίζα τον κόμβο xsd:short (εικόνα 5.16) και περιγράφουμε τη διαδικασία εύρεσης του βαθμού συμβατότητας μεταξύ του κόμβου-ρίζα και των κόμβων-παιδιών.



Εικόνα 5.16 Υποδέντρο με ρίζα τον xsd:short

Ξεκινάμε από το πρώτο επίπεδο του δέντρου στο οποίο βρίσκεται *μόνο* ο κόμβος xsd:int. Συνεπώς ο βαθμός συμβατότητας του ομώνυμου τύπου με τον xsd:short θα είναι ο μέγιστος δυνατός, δηλαδή ίσος με 1.0. Όσον αφορά τους κόμβους xsd:long και xsd:float που έχουν *κοινό γονέα*, ο βαθμός συμβατότητάς τους με τον xsd:short θα υπολογιστεί σύμφωνα με τη μετρική που εφαρμόστηκε στο παράδειγμα 1 και με το βαθμό συμβατότητας των κόμβων αυτών με τον xsd:int. Έχουμε λοιπόν: βαθμός συμβατότητας$_{xsd:long}$ = (1/1+1)·1.0· 0.7 = 0.35.

Ομοίως: βαθμός συμβατότητας$_{xsd:float}$ = (1/1+1)·1.0·0.3 = 0.15. Για τους υπόλοιπους κόμβους του δέντρου ακολουθείται η μετρική του παραδείγματος 1 από την οποία παίρνουμε τα αποτελέσματα που αναγράφονται στις ακμές του δέντρου.

Για να βρούμε το βαθμό συμβατότητας των υπολοίπων xml schema τύπων δεδομένων παίρνουμε τα αντίστοιχα υποδέντρα και εφαρμόζουμε τη διαδικασία που περιγράφτηκε

στα παραπάνω παραδείγματα. Η εικόνα 5.17 περιέχει τα αποτελέσματα αυτής της εφαρμογής σε κάθε ένα υποδέντρο.

xsd:int	
xsd:long	0.7
xsd:integer	0.35
xsd:float	0.3
xsd:double	0.15
xsd:decimal	0.1

xsd:byte	
xsd:short	1.0
xsd:int	0.5
xsd:long	0.233
xsd:integer	0.1725
xsd:float	0.1
xsd:double	0.075
xsd:decimal	0.06

xsd:short	
xsd:int	1.0
xsd:long	0.35
xsd:float	0.15
xsd:double	0.1
xsd:decimal	0.075
xsd:integer	0.233

xsd:float	
xsd:double	1.0
xsd:decimal	0.5

xsd:double	
xsd:decimal	1.0

xsd:integer	
xsd:decimal	1.0

xsd:long	
xsd:integer	1.0
xsd:decimal	0.5

xsd:unsignedShort	
xsd:unsignedInt	1.0
xsd:unsignedLong	0.5
xsd:integer	0.25
xsd:nonNegativeInteger	0.33
xsd:decimal	0.2

xsd:unsignedByte	
xsd:unsignedShort	1.0
xsd:unsignedInt	0.5
xsd:unsignedLong	0.33
xsd:integer	0.2
xsd:nonNegativeInteger	0.25
xsd:decimal	0.166

xsd:unsignedInt	
xsd:unsignedLong	1.0
xsd:integer	0.33
xsd:nonNegativeInteger	0.5
xsd:decimal	0.25

xsd:unsignedLong	
xsd:nonNegativeInteger	1.0
xsd:integer	0.5
xsd:decimal	0.33

xsd:positiveInteger	
xsd:nonNegativeInteger	1.0
xsd:integer	0.5
xsd:decimal	0.33

xsd:negativeInteger	
xsd:nonPositiveInteger	1.0
xsd:integer	0.5
xsd:decimal	0.33

xsd:nonNegativeInteger	
xsd:integer	1.0
xsd:decimal	0.5

xsd:nonPositiveInteger	
xsd:integer	1.0
xsd:decimal	0.5

Εικόνα 5.17 Βαθμοί συμβατότητας μεταξύ των xsd τύπων δεδομένων

5.1.2.2 Απεικόνιση γνωρισμάτων σε datatype-properties

Στην παρούσα ενότητα περιγράφεται ο αλγόριθμος απεικόνισης των γνωρισμάτων των σχέσεων της βάσης δεδομένων σε datatype-properties της οντολογίας. Όπως έχει ήδη αναφερθεί ο αλγόριθμος αυτός βασίζεται:

- Στο βαθμό συμβατότητας μεταξύ των τύπων δεδομένων των γνωρισμάτων και των datatype-properties.
- Στην «ομοιότητα» των ονομάτων τους.
- Στις αντιστοιχίσεις των πινάκων της βάσης σε κλάσεις της οντολογίας που προέκυψαν από τη φάση της αντιστοίχισης κλάσεων.

Στην περίπτωση της αντιστοίχισης των datatype-properties ο βαθμός (similarity) που «συνοδεύει» κάθε υποψήφιο datatype-property είναι μια συνάρτηση του βαθμού ομοιότητάς (name_similarity) του με το αντίστοιχο datatype-property της οντολογίας και του βαθμού συμβατότητας (compatibility) των τύπων δεδομένων τους. Η συνάρτηση αυτή έχει την ακόλουθη δομή:

similarity = w * name_similarity + (1 – w) * compatibility

όπου το w είναι μια σταθερά που παίρνει τιμή μέσα από το διάστημα [0,1].

Στην εικόνα 5.18 παρουσιάζεται ο αλγόριθμος αντιστοίχισης των γνωρισμάτων των σχέσεων της βάσης σε datatype-properties της οντολογίας.

Ο αλγόριθμος της εικόνας 5.18 προκειμένου να θεωρήσει το γνώρισμα a ενός πίνακα ως υποψήφιο datatype-property για ένα datatype-property dp της οντολογίας δεν ελέγχει μόνο αν ο πίνακας που περιέχει το a έχει αντιστοιχηθεί στο domain του dp, αλλά και την περίπτωση που ο πίνακας που περιέχει το a έχει αντιστοιχηθεί σε κάποια υπερ-κλάση του domain του dp.

```
Concepts = {all concepts of ontology}
For each concept c_i in Concepts do
  For each candidate concept c_i' in CCS_ci do
    Columns_Ci' = {all columns of candidate concept c_i'} \ {foreign keys and
                  auto-increment fields}
    DP_ci = {all datatype-properties with domain c_i or a super-concept of c_i}
    For each column a in Columns_Ci' do
      For each datatype-property dp in DP_ci do
```

```
            Compute the name_similarity between a and dp
            Compute the compatibility between datatype of a and the the range of dp
            If (datatype of a is equal to the the range of dp)
               then similarity = name_similarity
            else similarity = w * name_similarity + (1 – w) * compatibility
            If (similarity > threshold)
              then  CDPS_dp = CDPS_dp ∪ {a}
         End for
      End for
   End for
End for
```

Εικόνα 5.18 Αλγόριθμος αντιστοίχισης γνωρισμάτων σε datatype-properties.

Ας θεωρήσουμε για παράδειγμα τον πίνακα *WORKER* της εικόνας 5.5 και το τμήμα της οντολογίας της εικόνας 5.19. Σύμφωνα με την εικόνα 5.19 το datatype-property *hasName* έχει domain την κλάση *Person*. Επειδή όμως η κλάση *Person* είναι υπερ-κλάση της κλάσης *Worker* το *hasName* θεωρείται και datatype-property της κλάσης *Worker*. Αυτό συνεπάγεται πως αν ο πίνακας *WORKER* της εικόνας 5.5 έχει αντιστοιχηθεί στην κλάση Worker τότε είναι δυνατό το γνώρισμα *name* του *WORKER* να αποτελεί υποψήφιο datatype-property του *hasName*.

```
<owl:Class rdf:ID="Person"/>
 <owl:Class rdf:ID="Worker">
  <rdfs:subClassOf rdf:resource="#Person"/>
 </owl:Class>
 <owl:DatatypeProperty rdf:ID="hasName">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 </owl:DatatypeProperty>
```

Εικόνα 5.19 Ορισμός του datatype-property *hasName*

5.1.3 Αντιστοίχιση object-properties (Object-property mapping)

Ως γνωστόν τα object-properties εκφράζουν ένα είδος σχέσης ανάμεσα σε δύο κλάσεις της οντολογίας. Στη βάση δεδομένων οι σχέσεις (relationships) ανάμεσα σε δύο πίνακες

εκφράζονται μέσω των περιορισμών αναφορικής ακεραιότητας, δηλαδή με τη χρήση των ξένων κλειδιών. Εκτός όμως από τα ξένα κλειδιά υπάρχουν και πίνακες που εκφράζουν N:M σχέσεις ανάμεσα σε δύο άλλους πίνακες, όπως αναφέρθηκε και στην ενότητα 5.1.1, και οι οποίοι «αποκλείστηκαν» από τη φάση της αντιστοίχισης κλάσεων. Για να αποφανθούμε ότι ένας πίνακας T με σύνολο γνωρισμάτων A(T) εκφράζει μια N:M σχέση μεταξύ δύο άλλων πινάκων πρέπει να ικανοποιούνται οι ακόλουθες συνθήκες:

- Το σύνολο A(T) να περιέχει δύο γνωρίσματα A1 και A2 τέτοια ώστε το A1 $\cup$ A2 να αποτελεί το πρωτεύον κλειδί του T.
- Το A1 να είναι ξένο κλειδί προς κάποιον πίνακα T1.
- Το A2 να είναι ξένο κλειδί προς κάποιον πίνακα T2 (T1 $\neq$ T2).

Για παράδειγμα ο πίνακας EMPLOYMENT της εικόνας 5.5 είναι ένας πίνακας που εκφράζει μια N:M σχέση ανάμεσα στους πίνακες WORKER και WORK.

Αν ισχύουν οι παραπάνω συνθήκες τότε ο πίνακας T είναι δυνατόν να αποτελέσει υποψήφιο object-property για ένα object-property με domain D και range R αν ο πίνακας T1 έχει αντιστοιχηθεί στην κλάση D και ο πίνακας T2 έχει αντιστοιχηθεί στην κλάση R ή αν ο T1 έχει αντιστοιχηθεί στην R και ο T2 στην D.

```
OP = {all object properties of the ontology}
For each object property P in OP do
  D = domainOf(P)
  R = rangeOf(P)
  For each candidate concept C in CCS_D do
    FK = {foreignKeys of C}
    For each foreign key f in FK do
      If (f is "similar" to P AND the referenced table[1] of f ∈ CCS_R)
        then COPS_P = COPS_P ∪ {f}
    End for
  End for
End for
```

Εικόνα 5.20 Αλγόριθμος αντιστοίχισης ξένων κλειδιών σε object-properties

[1]referenced table: ο πίνακας στο πρωτεύον κλειδί του οποίου αναφέρεται κάποιο ξένο κλειδί

Ο αλγόριθμος της εικόνας 5.20 περιγράφει την διαδικασία αντιστοίχισης των περιορισμών αναφορικής ακεραιότητας σε object-properties εκμεταλλευόμενος την «ομοιότητα» των ονομάτων και τις υποψήφιες κλάσεις για κάθε κλάση τις οντολογίας που προέκυψαν από την αντιστοίχιση κλάσεων, ενώ ο αλγόριθμος της εικόνας 5.21 περιγράφει τη διαδικασία αντιστοίχισης των πινάκων της βάσης δεδομένων που αποτελούν N:M σχέσεις ανάμεσα σε δύο άλλους πίνακες, σε object-properties της οντολογίας.

Ο παραπάνω αλγόριθμος είναι γενικός και αφορά σε ξένα κλειδιά που εκφράζουν είτε 1:1 είτε 1:N σχέσεις ανάμεσα στους πίνακες που συνδέουν.

```
Relations = {all tables of the database that represent N:M relations}
Tables = {all tables of the database} \ Relations
OP = {all object properties of the ontology}
For each table t in Relations do
   ForegnKeys_t = {fk_1, fk_2} /*the two foreign keys of t that consist its primary key*/
   For each object property P in OP do
      D = domainOf(P)
      R = rangeOf(P)
      If ((D is mapped to the exported table of fk1 AND R is mapped to the exported
         table of fk2) OR (D is mapped to the referenced table of fk2 AND R is mapped
         to the exported table of fk1))
       then If (P is "similar" to t)
           then COP_P = COP_P ∪ {t}
    End for
End for
```

Εικόνα 5.21 Αλγόριθμος αντιστοίχισης πινάκων που αποτελούν N:M σχέσεις σε object-properties της οντολογίας.

5.2 Μετακίνηση δεδομένων (Data migration)

Από τη στιγμή που έχει ολοκληρωθεί η διαδικασία απεικόνισης του σχεσιακού μοντέλου στην οντολογία ξεκινά η μετακίνηση των δεδομένων από το ένα σχήμα στο δεύτερο, ο «εμπλουτισμός» δηλαδή της οντολογίας με δεδομένα προερχόμενα από τη βάση δεδομένων. Σε αντίθεση με την αντιστοίχιση σχήματος η διαδικασία αυτή γίνεται αυτόματα, χωρίς να καθίσταται απαραίτητη η παρέμβαση του χρήστη. Ο τρόπος με τον

οποίο επιτυγχάνεται αυτή η μετακίνηση των δεδομένων είναι έξω από το πλαίσιο της παρούσας εργασίας, ωστόσο στις ακόλουθες παραγράφους αναφέρονται κάποιες διαδικασίες που λαμβάνουν μέρος σ'αυτή τη φάση.

5.2.1 Μετατροπή των περιορισμών της οντολογίας σε SQL επερωτήσεις

Από τη στιγμή που το σχεσιακό σχήμα έχει αντιστοιχηθεί στην οντολογία γνωρίζουμε ποιοι πίνακες, γνωρίσματα και περιορισμοί αναφορικής ακεραιότητας έχουν αντιστοιχηθεί σε κλάσεις, datatype-properties και object-properties της οντολογίας. Αυτή η γνώση όμως δεν είναι αρκετή προκειμένου να αποφασίσουμε ποιες πλειάδες της βάσης θα «μετατραπούν» σε στιγμιότυπα (individuals) της οντολογίας. Ως γνωστόν το RONTO διαπραγματεύεται OWL οντολογίες οι οποίες μπορεί να περιέχουν περιορισμούς και αξιώματα. Είναι λοιπόν απαραίτητο να εξαχθούν από τη βάση δεδομένων εκείνες οι πλειάδες που δεν παραβιάζουν τους περιορισμούς της οντολογίας. Η εξαγωγή αυτή επιτυγχάνεται τις περισσότερες φορές με την αναπαράσταση των περιορισμών σε προτάσεις WHERE σε SQL επερωτήσεις.

5.2.2 Μετασχηματισμός των πεδίων της βάσης δεδομένων

Κατά τη διάρκεια της μετακίνησης των δεδομένων χειριζόμαστε τα πραγματικά δεδομένα της βάσης. Είναι δυνατόν κάποιες στήλες της βάσης να περιέχουν δεδομένα σε διαφορετική μορφή από αυτή που αναμένεται να έχουν τα datatype-properties της οντολογίας. Σε αυτές τις περιπτώσεις είναι απαραίτητη η παρέμβαση του χρήστη προκειμένου να οριστούν κάποιοι μετασχηματισμοί που πρέπει να γίνουν στις τιμές των πεδίων της βάσης πριν αυτά μετατραπούν σε στιγμιότυπα των αντίστοιχων στοιχείων της οντολογίας. Παραδείγματα τέτοιων μετασχηματισμών υπάρχουν στο [23].

5.2.3 Δημιουργία των στιγμιοτύπων της οντολογίας

Στη φάση αυτή εκτελούνται οι κανόνες που έχουν οριστεί για τη δημιουργία των στιγμιοτύπων της οντολογίας από τα δεδομένα της βάσης. Τα ονόματα των στιγμιοτύπων των κλάσεων της οντολογίας δημιουργούνται αυτόματα από τα πρωτεύοντα κλειδιά των υποψηφίων κλάσεων. Αξίζει να σημειωθεί ότι κατά τη διάρκεια εκτέλεσης του βήματος αυτού πρέπει να γίνει ειδική μεταχείριση των πεδίων της βάσης που περιέχουν NULL τιμές όπως επισημαίνεται στο [50].

5.3 Μέθοδοι υπολογισμού της ομοιότητας μεταξύ εννοιών

Στους αλγόριθμους που περιγράφτηκαν κατά την παρουσίαση της μεθοδολογίας με την οποία ένα σχεσιακό μοντέλο αντιστοιχίζεται σε μια οντολογία αναφέρθηκε πολλές φορές ο όρος «ομοιότητα». Για να αποφασιστεί, για παράδειγμα, αν ένας πίνακας της βάσης δεδομένων αποτελεί υποψήφια κλάση για μια συγκεκριμένη κλάση της οντολογίας έπρεπε πρώτα να ελεγχθεί αν το όνομα του πίνακα αυτού είναι «όμοιο» με το όνομα της αντίστοιχης κλάσης, έπρεπε δηλαδή να υπολογιστεί με κάποιο τρόπο η ομοιότητα μεταξύ δύο συμβολοσειρών. Δύο συμβολοσειρές όμως είναι όμοιες όχι μόνο αν έχουν στην ίδια θέση ίδιους χαρακτήρες αλλά ακόμα κι αν έχουν το ίδιο ακριβώς νόημα. Γι'αυτό λοιπόν ο υπολογισμός της ομοιότητας μεταξύ δύο συμβολοσειρών συνεπάγεται τη μέτρηση δύο διαφορετικών τύπων ομοιότητας, της γλωσσολογικής (ή γλωσσικής) (linguistic) και της σημασιολογικής (semantic) ομοιότητας. Στις ενότητες 5.2.1 και 5.2.2 περιγράφονται οι έννοιες της γλωσσολογικής και σημασιολογικής ομοιότητας που χρησιμοποιεί η μεθοδολογία που αναπτύχθηκε στην ενότητα 5.1 προκειμένου να αποφασίσει πότε ένα στοιχείο της βάσης δεδομένων μπορεί να αντιστοιχηθεί σε κάποιο στοιχείο της οντολογίας.

5.3.1 Γλωσσολογική ομοιότητα (Linguistic Similarity)

Ο υπολογισμός της γλωσσολογικής ομοιότητας μεταξύ δύο στοιχείων των δύο σχημάτων (π.χ. μεταξύ ενός πίνακα βάσης δεδομένων και μιας κλάσης της οντολογίας) βασίζεται στα ονόματα των στοιχείων αυτών.

Οι αλγόριθμοι εύρεσης της γλωσσολογικής ομοιότητας δύο συμβολοσειρών διακρίνονται σε αυτούς που υπολογίζουν την απόσταση των δύο συμβολοσειρών και σε αυτούς που υπολογίζουν το βαθμό ομοιότητάς τους. Τα αποτελέσματα των αλγορίθμων αυτών κανονικοποιούνται δίνοντας μια τιμή εντός του διαστήματος [0,1]. Η μέγιστη τιμή 1 δηλώνει ότι οι δύο συμβολοσειρές είναι πανομοιότυπες, ενώ η τιμή 0 υποδηλώνει ότι οι δύο συγκρινόμενες συμβολοσειρές δεν έχουν κανένα κοινό στοιχείο.

Η απόσταση συμβολοσειρών (edit distance) είναι μια μετρική, η οποία διατυπώθηκε από τον Levenshtein [25], που υπολογίζει τον ελάχιστο αριθμό των εισαγωγών, διαγραφών και αντικαταστάσεων που πρέπει να γίνουν ώστε η πρώτη συμβολοσειρά να μετατραπεί στη δεύτερη. Για παράδειγμα οι συμβολοσειρές "car" και "cat" έχουν edit distance 1. Μερικοί αλγόριθμοι εύρεσης της ελάχιστης απόστασης δύο συμβολοσειρών είναι: η απόσταση Hamming, ο αλγόριθμος Jaro [26], ο αλγόριθμος Needleman-Wunch

[27], η τεχνική q-gram [28], κ.ά. Για παράδειγμα, σύμφωνα με τη μετρική Jaro η τιμή της ομοιότητας δύο συμβολοσειρών $s = s_1s_2\ldots s_k$ και $t = t_1t_2\ldots t_m$ υπολογίζεται με βάση τον τύπο:

$$\text{Jaro}(s,t) = \frac{1}{3}\left(\frac{|s'|}{|s|}+\frac{|t'|}{|t|}+\frac{|s'|-T_{s',t'}}{|s'|}\right)$$

όπου s' είναι η ακολουθία χαρακτήρων της συμβολοσειράς s που είναι κοινοί με χαρακτήρες της συμβολοσειράς t, t' είναι η ακολουθία χαρακτήρων της συμβολοσειράς t που είναι κοινοί με χαρακτήρες της συμβολοσειράς s, $T_{s',t'}$ είναι το μισό του πλήθους των μετασχηματισμών των s' και t', όπου ένας μετασχηματισμός είναι μια παραλλαγή της ακολουθίας των s' και t' ώστε σε κάθε θέση i να υπάρχουν χαρακτήρες διαφορετικοί μεταξύ τους και τέλος ο συμβολισμός |x| παριστάνει το μήκος της συμβολοσειράς x. Θεωρούμε ότι ένας χαρακτήρας s_i της συμβολοσειράς s είναι κοινός με κάποιον χαρακτήρα της συμβολοσειράς t αν υπάρχει $t_j = s_i$ τέτοιο ώστε $i - H \leq j \leq i + H$, με $H=\frac{\min(|s|,|t|)}{2}$.

Οι αλγόριθμοι της δεύτερης κατηγορίας, υπολογίζουν το βαθμό ομοιότητας μεταξύ δύο συμβολοσειρών είτε κανονικοποιώντας την τιμή που δίνει η ελάχιστη απόσταση στο διάστημα [0,1], συνήθως διαιρώντας την με τη μέγιστη τιμή, είτε χρησιμοποιώντας την τιμή της ελάχιστης απόστασης σε κάποιο τύπο εξάγοντας το τελικό αποτέλεσμα. Παραδείγματα τέτοιων αλγορίθμων είναι: ο συντελεστής Dice, ο συντελεστής Lin [30], κ.ά.

5.3.2 Σημασιολογική Ομοιότητα (Semantic Similarity)

Χρησιμοποιώντας μόνο μεθόδους εύρεσης της γλωσσολογικής ομοιότητας μεταξύ δύο εννοιών είναι πολύ πιθανό να μην εξάγουμε πάντα τα επιθυμητά αποτελέσματα. Υπάρχουν περιπτώσεις όπου ενώ η σύγκριση της γλωσσολογικής ομοιότητας ανάμεσα σε δύο έννοιες, όπως μεταξύ των λέξεων “car” και “cat” που αναφέρθηκαν και στην προηγούμενη ενότητα, οδηγεί σε ένα σχετικά μεγάλο βαθμό (π.χ., 0,77 με τη χρήση της μετρικής Jaro), στην πραγματικότητα οι δύο έννοιες δεν σχετίζονται με κανέναν τρόπο αφού έχουν εντελώς διαφορετική σημασιολογία. Υπάρχουν επίσης και έννοιες οι οποίες ενώ δεν είναι γλωσσολογικά όμοιες, σχετίζονται με κάποιο τρόπο. Για παράδειγμα, οι λέξεις “Worker” και “Employee” έχουν μικρό βαθμό γλωσσολογικής ομοιότητας (0,18 κατά Jaro), ωστόσο η αντιστοίχιση του πίνακα “Worker” στην κλάση “Employee” θα

ήταν εφικτή αφού οι δύο λέξεις είναι συνώνυμες. Καθίσταται λοιπόν αναγκαίος ο υπολογισμός και ενός άλλου είδους ομοιότητας μεταξύ δύο εννοιών, της σημασιολογικής ομοιότητας. Η ανάγκη υπολογισμού της σημασιολογικής ομοιότητας ενισχύεται από το γεγονός ότι τα προς αντιστοίχιση σχήματα (σχεσιακό και οντολογία) είναι πιθανόν να έχουν σχεδιαστεί από διαφορετικά άτομα και ως εκ τούτου διαφορετικές λέξεις μπορεί να έχουν χρησιμοποιηθεί για την αναπαράσταση της ίδιας έννοιας.

Οι αλγόριθμοι υπολογισμού του βαθμού της σημασιολογικής ομοιότητας μεταξύ δύο εννοιών που χρησιμοποιούνται από την παρούσα μεθοδολογία για την αντιστοίχιση των στοιχείων του σχεσιακού σχήματος στα στοιχεία της οντολογίας κάνουν χρήση του ηλεκτρονικού λεξικού WordNet που κατασκευάστηκε από το πανεπιστήμιο του Princeton [32].

Το WordNet [33] είναι ένα σημασιολογικό λεξικό για την Αγγλική γλώσσα. Το λεξικό αυτό ομαδοποιεί τις λέξεις σε σύνολα συνωνύμων, τα αποκαλούμενα synsets, ενώ ταυτόχρονα καταγράφει τις σημασιολογικές σχέσεις που εμφανίζονται ανάμεσα στα παραπάνω σύνολα. Κάθε σύνολο συνωνύμων αναπαριστά μια διακριτή έννοια (sense). Για παράδειγμα στην εικόνα 5.22 φαίνονται οι έννοιες της λέξης “child”. Το WordNet διαχωρίζει ουσιαστικά, ρήματα, επίθετα και επιρρήματα με την υπόθεση ότι τα συγκεκριμένα μέρη του λόγου επεξεργάζονται διαφορετικά από τον ανθρώπινο εγκέφαλο. Κάθε synset περιέχει ένα σύνολο από συνώνυμες λέξεις ή παραθέσεις (σύνολα από λέξεις που χρησιμοποιούνται μαζί ώστε να αποδώσουν ένα συγκεκριμένο νόημα). Συνήθως μία λέξη περιέχεται σε περισσότερα του ενός synsets.

Κάθε synset συνδέεται με άλλα synsets μέσω συγκεκριμένων συσχετισμών. Οι συσχετισμοί αυτοί ορίζονται ανάλογα με το είδος της ευρετηριαζόμενης λέξης. Έτσι, ανάλογα με το τι μέρος του λόγου είναι η κάθε λέξη διακρίνονται οι ακόλουθες κατηγορίες:

- Ουσιαστικά:
    - Συνώνυμα: synsets με παρεμφερές νόημα
    - Υπέρνυμα: ένα synset Y θεωρείται υπέρνυμο του X αν κάθε έννοια του X είναι συναφής με το Y
    - Υπόνυμα: ένα synset Y θεωρείται υπόνυμο του X αν κάθε έννοια του Y είναι συναφής με το X
    - Συναφή: Δύο synsets είναι συναφή αν εμφανίζουν ένα κοινό υπέρνυμο.

- Ολόνυμα: ένα synset Y είναι ολόνυμο του X αν το X είναι τμήμα του Y
- Μερώνυμα: ένα synset Y είναι μερώνυμο του X αν το Y είναι τμήμα του X

- Ρήματα
  - Συνώνυμα: synsets με παρεμφερές νόημα
  - Υπέρνυμα: ένα synset Y θεωρείται υπέρνυμο του X αν κάθε έννοια του X είναι συναφής με το Y
  - Συναφή: Δύο synsets είναι συναφή αν εμφανίζουν ένα κοινό υπέρνυμο.

- Επίθετα:
  - Συνώνυμα: synsets (επιθέτων ή ουσιαστικών) με παρεμφερές νόημα
  - Αντώνυμα: synsets με διαφορετικό νόημα

- Επιρρήματα:
  - Συνώνυμα: synsets με παρεμφερές νόημα
  - Αντώνυμα: synsets με διαφορετικό νόημα

Το WordNet παρέχει επίσης και την έννοια του επιπέδου πολυσημίας. Το επίπεδο πολυσημίας μιας λέξης αντιστοιχεί στον αριθμό των synsets που περιέχουν τη λέξη αυτή. Αν κάποια λέξη συμμετέχει σε πολλά synsets, τότε τα τελευταία μπορούν να ιεραρχηθούν με βάση τη συχνότητα εμφάνισής τους. Η συχνότητα ορίζεται από την επεξεργασία κειμένων που περιέχουν την εξεταζόμενη λέξη σε διαφορετικά εννοιολογικά περιβάλλοντα.

Μερικοί αλγόριθμοι εύρεσης της σημασιολογικής ομοιότητας μεταξύ δύο εννοιών, οι οποίοι κάνουν χρήση του συγκεκριμένου λεξικού είναι: ο αλγόριθμος Wu-Palmer [35], ο αλγόριθμος Leacock-Chodorow [34], κ.ά. Περισσότερες πληροφορίες για τους αλγορίθμους εύρεσης τόσο της γλωσσολογικής όσο και της σημασιολογικής ομοιότητας δύο εννοιών υπάρχουν στο [51].

The noun child has 4 senses (first 4 from tagged texts)

1. (625) *child*, kid, youngster, minor, shaver, nipper, small fry, tiddler, tike, tyke, fry, nestling -- (a young person of either sex; "she writes books for children"; "they're just kids"; "`tiddler' is a British term for youngsters")
2. (186) *child*, kid -- (a human offspring (son or daughter) of any age; "they had three children"; "they were able to send their kids to college")3. (9) *child*, baby -- (an immature childish person; "he remained a child in practical matters as long as he lived"; "stop being a baby!")
4. (3) *child* -- (a member of a clan or tribe; "the children of Israel")

Εικόνα 5.22 Οι έννοιες της λέξης “child” σύμφωνα με το WordNet.

5.4 Σύνοψη

Στην ενότητα αυτή συνοψίζουμε - στον πίνακα 5.3 - τα βήματα της μεθοδολογίας που ακολουθεί το RONTO με τη σειρά με την οποία εκτελούνται. Η τελευταία στήλη του πίνακα δηλώνει αν η αντίστοιχη διαδικασία γίνεται αυτόματα (automatic), με τη βοήθεια του χρήστη (semi-automatic) ή εντελώς χειροκίνητα (manually).

Πίνακας 5.3 Η μεθοδολογία του RONTO

Βήματα της μεθοδολογίας του RONTO	**Τρόπος εκτέλεσης**
Αντιστοίχιση σχημάτων (Schema mapping)	
1) Απλή αντιστοίχιση κλάσεων	ημι-αυτόματα
2) Αντιστοίχιση datatype-properties και object-properties για τα CCSs που προέκυψαν από το βήμα 1	ημι-αυτόματα
3) Σύνθετη αντιστοίχιση κλάσεων	ημι-αυτόματα
4) Αντιστοίχιση datatype-properties και object-properties για τα CCSs που προέκυψαν από το βήμα 2	ημι-αυτόματα
Μετακίνηση δεδομένων (Data migration)	
1) Μετατροπή των περιορισμών της οντολογίας σε SQL επερωτήσεις	ημι-αυτόματα
2) Μετασχηματισμός των πεδίων της βάσης δεδομένων	χειροκίνητα
3) Δημιουργία των στιγμιοτύπων της οντολογίας	αυτόματα

ΚΕΦΑΛΑΙΟ 6

ΑΞΙΟΛΟΓΗΣΗ

Στο κεφάλαιο αυτό παρουσιάζονται η απόδοση και η λειτουργικότητα κάποιων αλγορίθμων που περιγράφτηκαν στην ενότητα 5.1. Συγκεκριμένα οι αλγόριθμοι που υλοποιούν την απλή αντιστοίχιση κλάσεων (ενότητα 5.1.1.1) και την αντιστοίχιση των datatype-properties (ενότητα 5.1.2) δοκιμάστηκαν σε τέσσερα σύνολα δεδομένων από διαφορετικές θεματικές περιοχές και στο παρόν κεφάλαιο αναλύονται τα αποτελέσματά τους.

Στην ενότητα 6.1 δίνεται μια περιγραφή των συνόλων δεδομένων που χρησιμοποιήθηκαν για την τεκμηρίωση της παρούσας μελέτης ενώ στην ενότητα 6.2 παρουσιάζονται τα κριτήρια που λήφθηκαν υπόψιν προκειμένου να εξαχθούν συμπεράσματα για την απόδοση της προτεινόμενης μεθοδολογίας. Τέλος, στις ενότητες 6.2 και 6.3 παρουσιάζονται και αναλύονται τα αποτελέσματα του concept mapping και του datatype-property mapping αντίστοιχα.

6.1 Περιγραφή συνόλων δεδομένων

Στην ενότητα αυτή περιγράφονται συνοπτικά τέσσερα σχήματα σχεσιακών βάσεων δεδομένων και τέσσερις οντολογίες που χρησιμοποιήθηκαν προκειμένου να αξιολογηθεί μέρος της προτεινόμενης μεθοδολογίας. Αν και το πλήθος των στοιχείων που συνιστούν τα τέσσερα αυτά σύνολα δεδομένων δεν είναι αρκετά μεγάλο, τα όνοματά τους έχουν αρκετές διαφορές δεδομένου ότι κατασκευάστηκαν από διαφορετικά άτομα. Στο παράρτημα Α υπάρχει αναλυτική παρουσίασή τους.

- *1^{ο} Σύνολο δεδομένων "COMPANY" :* Το συγκεκριμένο σύνολο δεδομένων περιέχει πληροφορίες σχετικά με τα στοιχεία εκείνα που συνιστούν μια εταιρεία,

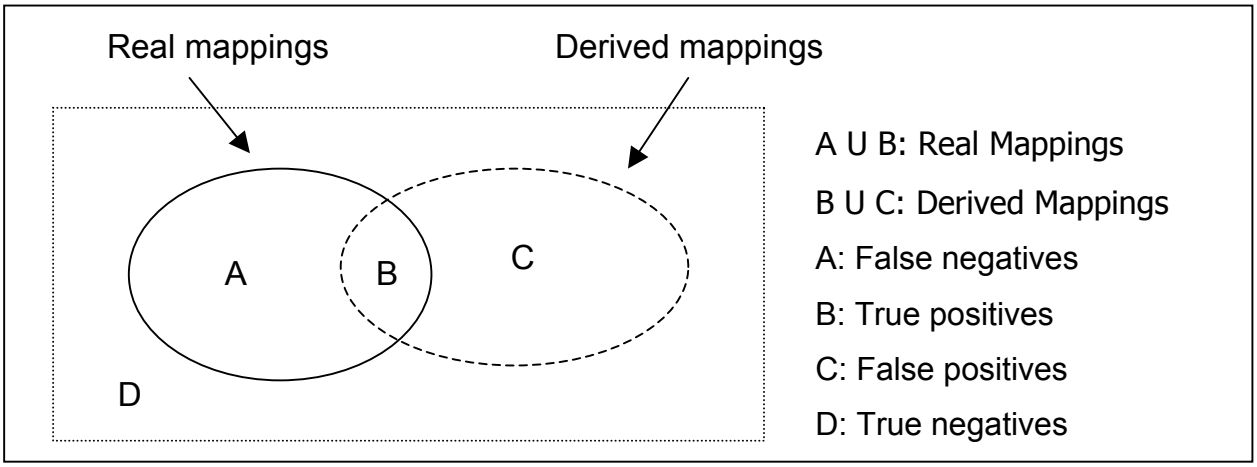
όπως για παράδειγμα οι εργαζόμενοι που δουλεύουν σε αυτή, τα τμήματα που την αποτελούν κτλ.

- *2ο Σύνολο δεδομένων "ISWC – International Semantic Web Conference" :* Σ'αυτό το σύνολο δεδομένων περιέχονται πληροφορίες σχετικά με τα άτομα, τους οργανισμούς και τα άρθρα που πήραν μέρος στο Διεθνές Συνέδριο του Σημασιολογικού Ιστού.
- *3ο Σύνολο δεδομένων "LIBRARIES" :* Στο σύνολο δεδομένων με όνομα "LIBRARIES" περιέχονται πληροφορίες σχετικά με τα βιβλία, τους συγγραφείς αυτών και τις τοποθεσίες κάποιων βιβλιοθηκών.
- *4ο Σύνολο δεδομένων "PERSONS" :* Το συγκεκριμένο σύνολο δεδομένων περιέχει πληροφορίες για άτομα που δουλεύουν και σπουδάζουν σε επιχειρήσεις και πανεπιστήμια αντίστοιχα.

6.2 Μετρικές αξιολόγησης

Προκειμένου να αξιολογήσουμε τα αποτελέσματα του αλγορίθμου της απλής αντιστοίχισης κλάσεων και του αλγορίθμου που αντιστοιχεί τα πεδία των πινάκων της βάσης δεδομένων σε datatype-properties της οντολογίας εισάγουμε ορισμένες μετρικές [37] που μετρούν την ποιότητα των αντιστοιχίσεων (match quality metrics). Πριν εξάγουμε τιμές για κάθε μία από αυτές τις μετρικές θα πρέπει να βρούμε (χειρωνακτικώς) τις ακριβείς – πραγματικές - αντιστοιχίσεις (real mappings) ανάμεσα στο σχεσιακό και στο εννοιολογικό μοντέλο και στη συνέχεια να συγκρίνουμε τις αντιστοιχίσεις που θα μας δώσουν αυτόματα οι αλγόριθμοι (derived mappings) με τις πραγματικές. Οι πραγματικές αντιστοιχίσεις των συνόλων δεδομένων που χρησιμοποιούνται στο κεφάλαιο αυτό βρίσκονται στο Παράρτημα Β και γίνονται συνήθως από ειδικούς (domain experts).

Παρακάτω αναλύονται οι μετρικές που χρησιμοποιούμε προκειμένου να μετρήσουμε την απόδοση των αλγορίθμων.



Εικόνα 6.1 Σύγκριση μεταξύ των real mappings και των derived mappings

Στην εικόνα 6.1 συγκρίνεται το σύνολο των πραγματικών αντιστοιχίσεων με τις αντιστοιχίσεις που δίνουν αυτόματα οι αλγόριθμοι. Σύμφωνα με την εικόνα 6.1:

- το σύνολο A αναπαριστά όλες τις πραγματικές αντιστοιχίσεις που δεν έδωσαν οι αλγόριθμοι (false negatives),
- το σύνολο B αναπαριστά όλες τις σωστές αντιστοιχίσεις που έδωσαν οι αλγόριθμοι (true positives),
- το σύνολο C αναπαριστά όλες τις αντιστοιχίσεις που έδωσαν οι αλγόριθμοι αλλά δεν είναι σωστές (false positives) και
- το σύνολο D αναπαριστά όλες τις λανθασμένες αντιστοιχίσεις που σωστά απέρριψαν οι αλγόριθμοι (true negatives).

Με βάση τον αριθμό των στοιχείων (cardinality) των παραπάνω συνόλων ορίζονται τα ακόλουθα μεγέθη:

- **Precision = |B| / (|B| + |C|)** $\longrightarrow$ αναπαριστά το ποσοστό των πραγματικών αντιστοιχίσεων που έδωσε ο αλγόριθμος σε σχέση με όλες τις αντιστοιχίσεις που έδωσε ο αλγόριθμος.
- **Recall = |B| / (|A| + |B|)** $\longrightarrow$ αναπαριστά το μερίδιο των πραγματικών αντιστοιχίσεων που έδωσε ο αλγόριθμος σε σχέση με όλες τις πραγματικές αντιστοιχίσεις.
- **F-measure = 2 * Precision * Recall / (Precision + Recall)** $\longrightarrow$ αναπαριστά τον αρμονικό μέσο όρο του Precision και Recall.

- **Overall = Recall * (2 – 1/Precision)** ⟶ προτείνεται στο [24] και χρησιμοποιήθηκε και στο [20] για να ποσοτικοποιεί την προσπάθεια που απαιτείται προκειμένου να προστεθούν στις αντιστοιχίσεις τα στοιχεία του συνόλου A και να αφαιρεθούν τα στοιχεία του συνόλου C.

Στην ιδανική περίπτωση, όπου το πλήθος των στοιχείων του συνόλου A και C είναι μηδενικό, έχουμε: Precision = Recall = 1. Όσο οι τιμές των παραπάνω μεγεθών πλησιάζουν το 1 τόσο πιο αποδοτική είναι και η διαδικασία της αντιστοίχισης των δύο σχημάτων. Παρόλα αυτά ούτε το Precision ούτε το Recall μπορούν να αποτιμήσουν την ποιότητα της αντιστοίχισης [37]. Συγκεκριμένα, το Recall μπορεί να αυξηθεί σε βάρος του Precision χρησιμοποιώντας μικρή τιμή για το threshold (το κατώτατο όριο με βάση το οποίο αποφασίζεται αν ένα στοιχείο του σχεσιακού σχήματος μπορεί να αντιστοιχηθεί σε κάποιο στοιχείο της οντολογίας) με αποτέλεσμα να αυξάνεται και το πλήθος των αντιστοιχίσεων. Από την άλλη μεριά, αν στο αποτέλεσμα εμφανίζονται μόνο λίγες αλλά πραγματικές αντιστοιχίσεις τότε επιτυγχάνουμε μεγάλη τιμή για το Precision αλλά μικρή για το Recall.

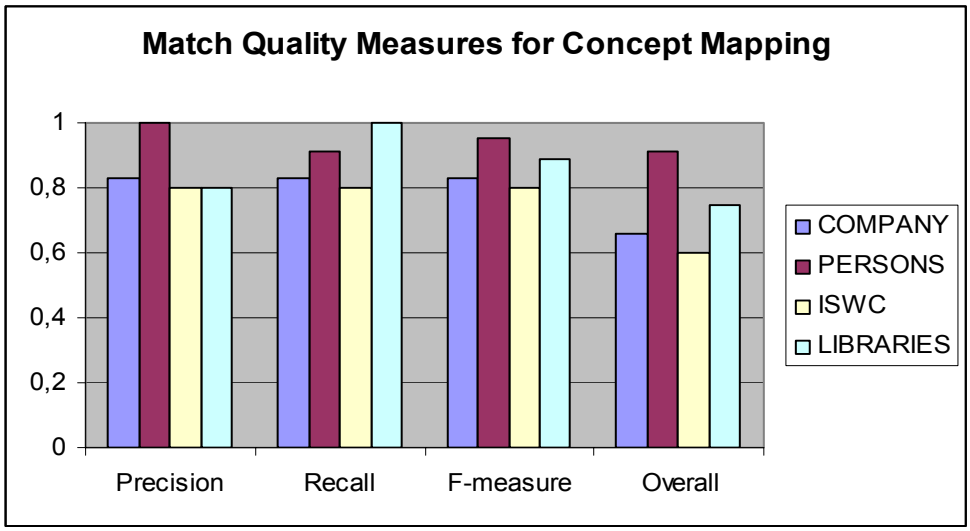
Στις παραγράφους 6.3 και 6.4 αποτιμώνται τα μεγέθη αυτά για τις διαδικασίες της απλής αντιστοίχισης κλάσεων και της αντιστοίχισης των datatype-properties αντίστοιχα με στόχο την επιλογή του κατάλληλου threshold.

6.3 Αξιολόγηση της απλής αντιστοίχισης κλάσεων με το κατάλληλο threshold

Ο αλγόριθμος της απλής αντιστοίχισης κλάσεων που περιγράφτηκε στην ενότητα 5.1.1.1 εφαρμόστηκε για τα τέσσερα προαναφερθέντα ζεύγη σχημάτων και έδωσε τα αποτελέσματα του σχήματος 6.1. Στο Παράρτημα Γ δίνονται όλες οι αντιστοιχίσεις που έδωσε ο αλγόριθμος και για τα τέσσερα ζεύγη σχημάτων.

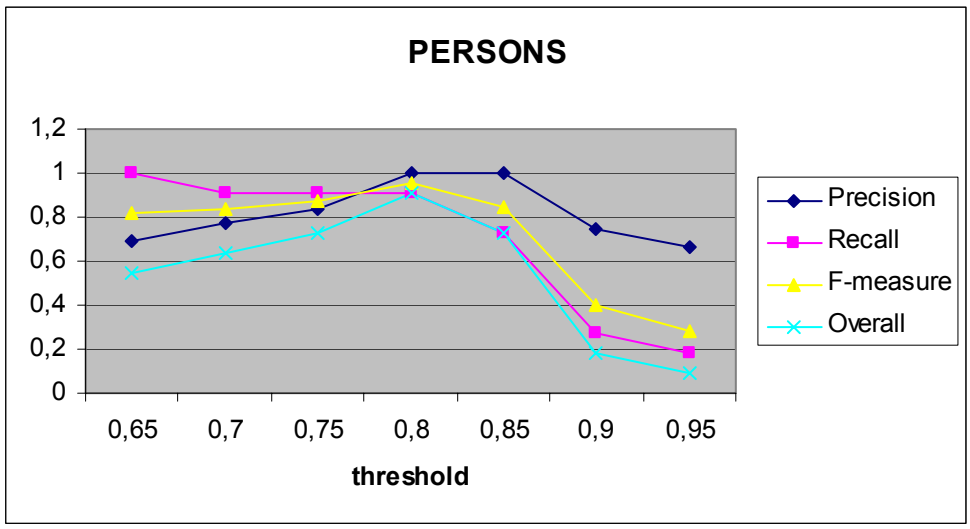
Για την εύρεση των αντιστοιχίσεων ανάμεσα στους πίνακες της βάσης δεδομένων και στις κλάσεις της οντολογίας δόθηκε μεγάλη βαρύτητα στη σημασιολογική ομοιότητα των αντίστοιχων ονομάτων αφού ο επιδιωκόμενος στόχος είναι η εύρεση σημασιολογικά όμοιων στοιχείων ανάμεσα στα σχεσιακό και εννοιολογικό σχήμα. Στις περιπτώσεις που τα ονόματα των προς αντιστοίχιση πινάκων και κλάσεων δεν ήταν έγκυρα (π.χ πίνακας *bindex* στο σχεσιακό σχήμα *LIBRARIES*) η εύρεση του βαθμού ομοιότητας εστιάστηκε μόνο σε μεθόδους μέτρησης της γλωσσολογικής ομοιότητας.

Παρατηρούμε ότι ο αλγόριθμος έδωσε πολύ καλά αποτελέσματα και για μεγάλα σύνολα δεδομένων όπως είναι το “PERSONS” αλλά και για τα σχήματα που είχαν σημαντικές διαφορές στα ονόματα των στοιχείων τους όπως είναι το “COMPANY”.

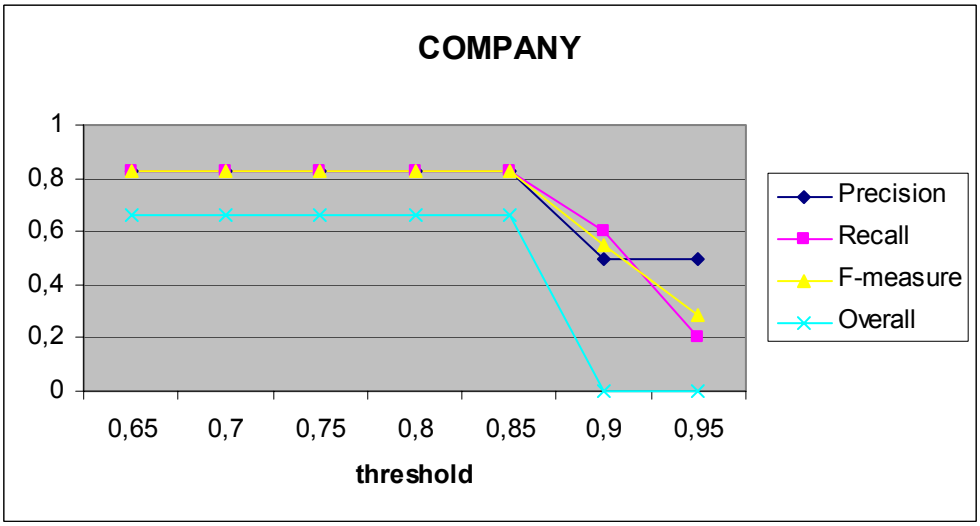


Σχήμα 6.1 Συγκεντρωτικά αποτελέσματα της απλής αντιστοίχισης κλάσεων. Με τι threshold?

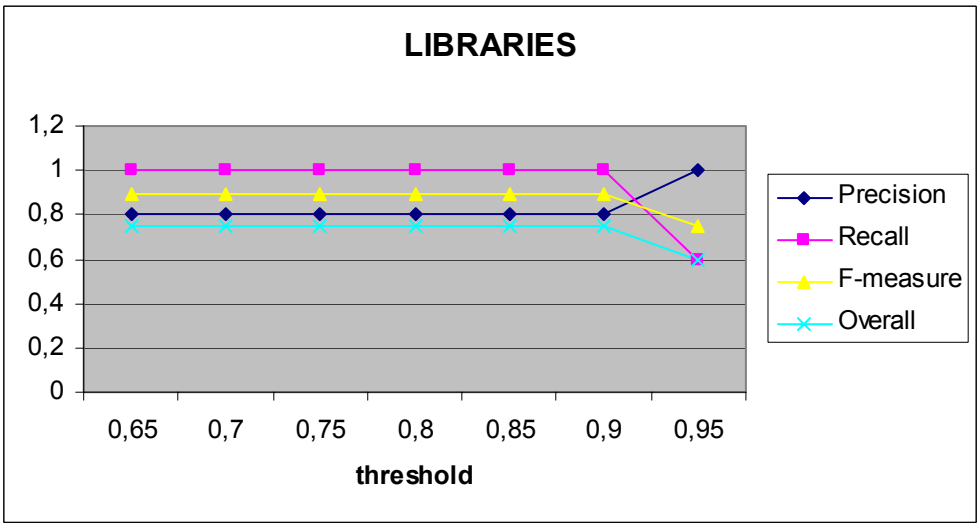
Για να αποφασίσει ο αλγόριθμος πότε ένας πίνακας της οντολογίας αποτελεί υποψήφια κλάση για μια συγκεκριμένη κλάση της οντολογίας πρέπει πρώτα να μετρήσει την ομοιότητα των ονομάτων τους και να ελέγξει αν ο βαθμός ομοιότητας ξεπερνάει ένα κατώτατο όριο (threshold). Το όριο αυτό εξαρτάται από τα εκάστοτε σχήματα (σχεσιακό και εννοιολογικό) και μπορεί να καθοριστεί από το χρήστη. Στα σχήματα 6.2, 6.3, 6.4 και 6.5 υπολογίστηκαν τα μεγέθη που χρησιμοποιούνται στην αξιολόγηση για κάθε ένα από τα σύνολα δεδομένων του παραρτήματος Α προκειμένου να αποφασιστεί το πιο κατάλληλο κατώτατο όριο. Το πιο κατάλληλο λοιπόν threshold είναι αυτό που δίνει τη μέγιστη τιμή στο μέγεθος F-measure αφού στο σημείο αυτό το Precision και το Recall έχουν την ελάχιστη απόσταση μεταξύ τους.



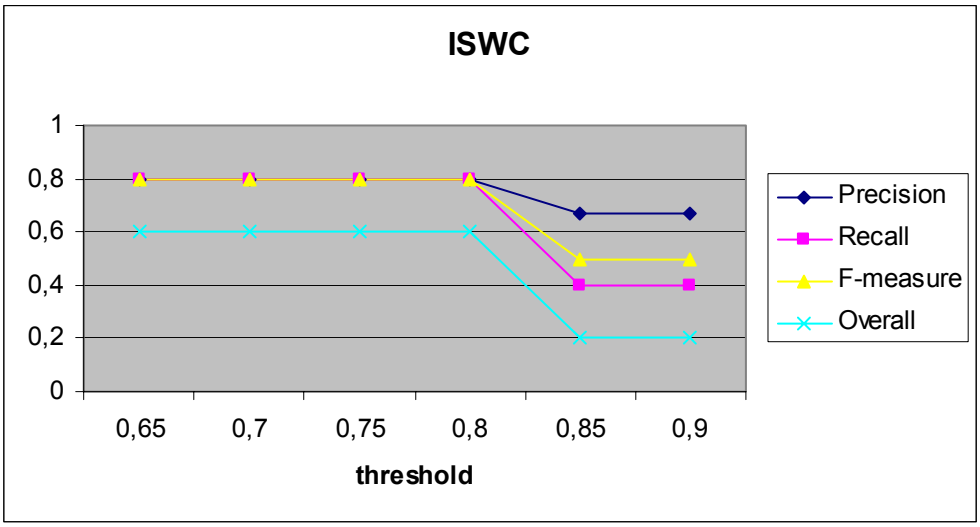
Σχήμα 6.2 Αποτελέσματα απλής αντιστοίχισης κλάσεων για το σύνολο δεδομένων "PERSONS"



Σχήμα 6.3 Αποτελέσματα απλής αντιστοίχισης κλάσεων για το σύνολο δεδομένων "COMPANY"



Σχήμα 6.4 Αποτελέσματα απλής αντιστοίχισης κλάσεων για το σύνολο δεδομένων "LIBRARIES"



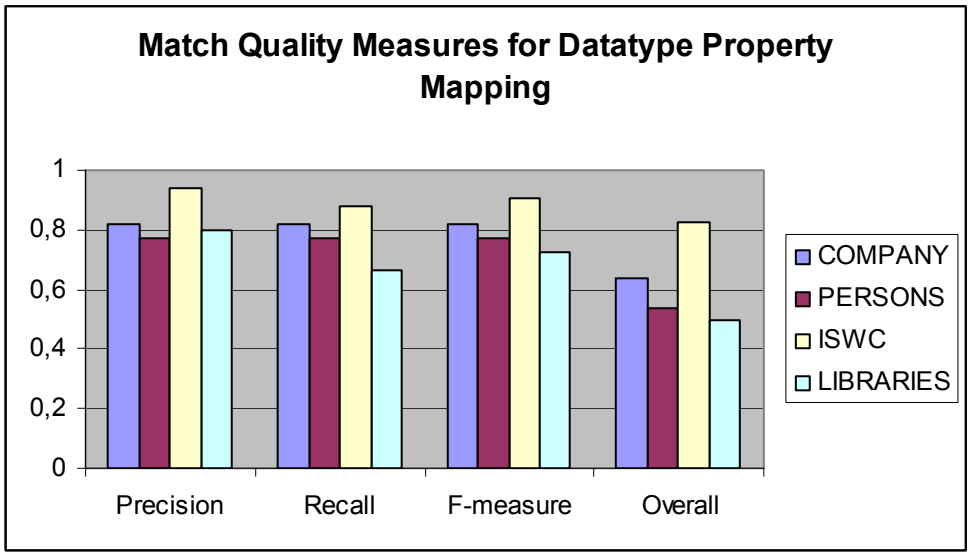
Σχήμα 6.5 Αποτελέσματα απλής αντιστοίχισης κλάσεων για το σύνολο δεδομένων "ISWC"

Σύμφωνα με τις γραφικές παραστάσεις των παραπάνω σχημάτων οι καλύτερες τιμές για το threshold είναι οι ακόλουθες:

- PERSONS → 0,8
- COMPANY → 0,85
- ISWC → 0,8
- LIBRARIES → 0,9

6.4 Αξιολόγηση της αντιστοίχισης των πεδίων σε datatype-properties με το κατάλληλο threshold

Όπως έχει αναφερθεί, την αντιστοίχιση των πινάκων σε κλάσεις της οντολογίας διαδέχεται η διαδικασία αντιστοίχισης των πεδίων των πινάκων σε datatype-properties της οντολογίας. Στην ενότητα αυτή αξιολογούνται τα αποτελέσματα του αλγορίθμου της ενότητας 5.1.2.2. Ο αλγόριθμος αυτός εφαρμόστηκε στα σύνολα δεδομένων του παραρτήματος Α και έδωσε τα αποτελέσματα του σχήματος 6.6.



Σχήμα 6.6 Συγκεντρωτικά αποτελέσματα της αντιστοίχισης των πεδίων σε datatype-properties της οντολογίας.

Σημαντικό ρόλο στην αντιστοίχιση των πεδίων των πινάκων σε datatype-properties της οντολογίας δεν παίζει μόνο η ομοιότητα των ονομάτων τους αλλά και ο βαθμός συμβατότητας των αντίστοιχων xsd τύπων δεδομένων τους. Στον υπολογισμό λοιπόν της ομοιότητας που υφίσταται ανάμεσα σε ένα πεδίο ενός πίνακα και σε ένα datatype-property μιας οντολογίας συμμετέχουν ο βαθμός ομοιότητας (name similarity) των ονομάτων τους (σημασιολογικός και γλωσσικός) και ο βαθμός συμβατότητας των τύπων δεδομένων τους (datatype compatibility). Κάθε βαθμός όμως δεν έχει την ίδια βαρύτητα στον υπολογισμό του τελικού βαθμού ομοιότητας (similarity). Ο βαθμός ομοιότητας των ονομάτων γενικά παίζει το σημαντικότερο ρόλο αφού η συμβατότητα των τύπων δεδομένων μπορεί να επηρεαστεί από το πόσο καλά σχεδιασμένα είναι τα προς αντιστοίχιση σχήματα. Ας θεωρήσουμε, για παράδειγμα, το πεδίο *salary* του πίνακα *worker* της βάσης δεδομένων *COMPANY* και το datatype-property *hasSalary* της αντίστοιχης οντολογίας και ας υποθέσουμε ότι ο τύπος δεδομένων του *salary* είναι xsd:integer ενώ ο τύπος δεδομένων του *hasSalary* είναι xsd:positiveInteger. Ο xsd:positiveInteger δεν είναι συμβατός με τον xsd:integer, ωστόσο μια αντιστοίχιση ανάμεσα στο πεδίο *salary* και στο datatype-property *hasSalary* είναι εφικτή αφού η ομοιότητα των ονομάτων τους είναι αρκετά μεγάλη και η διαφορά των τύπων δεδομένων οφείλεται είτε στην αδυναμία εξαγωγής περιορισμών από τη βάση δεδομένων σε περίπτωση που έχουν οριστεί, ή στον «κακό» σχεδιασμό της βάσης δεδομένων σε περίπτωση που δεν έχει δηλωθεί ότι οι τιμές του πεδίου *salary* πρέπει να

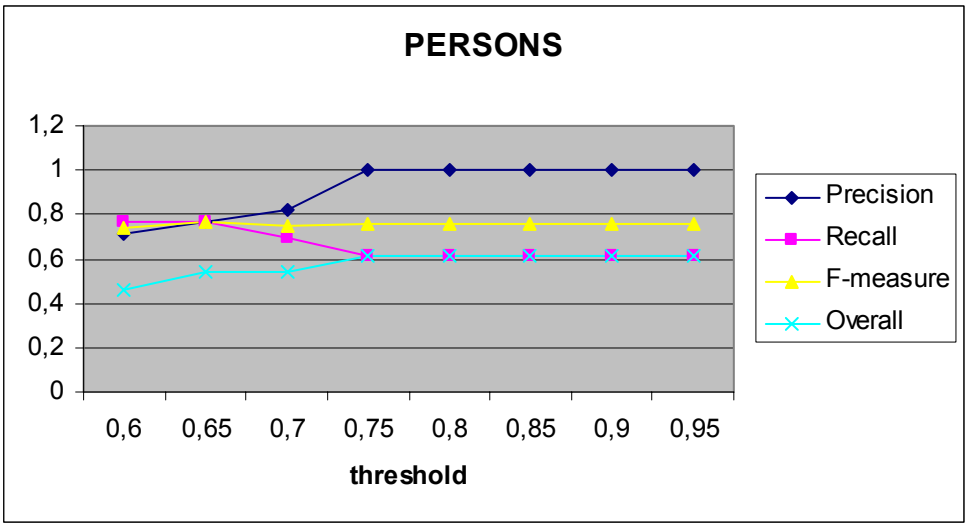
είναι θετικές. Σύμφωνα με πειράματα που έγιναν στα σύνολα δεδομένων του παραρτήματος Α ο κατάλληλος τρόπος για τον υπολογισμό του τελικού βαθμού ομοιότητας ενός πεδίου και ενός datatype-property είναι με βάση την ακόλουθη σχέση:

similarity = 0,8 * name_similarity + 0,2 * datatype compatibility

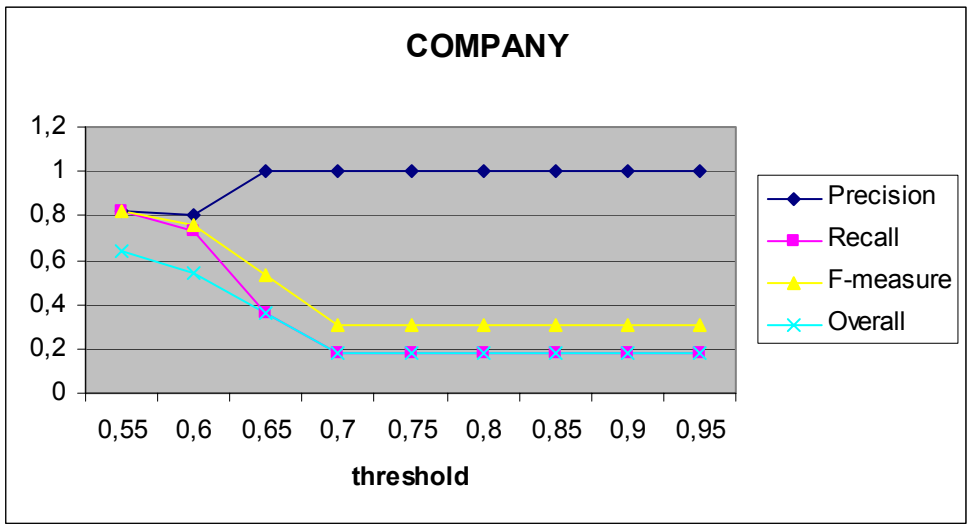
Ανάλογα με την αντιστοίχιση των πινάκων σε κλάσεις της οντολογίας έτσι και στην αντιστοίχιση των πεδίων των πινάκων σε datatype-properties της οντολογίας ορίστηκε ένα κατώτατο όριο βάσει του οποίου ο αλγόριθμος βρίσκει υποψήφια datatype-properties για κάθε datatype-property της οντολογίας. Το κατάλληλο όριο για κάθε ζεύγος σχημάτων καθορίστηκε με βάση τη μέγιστη τιμή της F-measure όπως δείχνουν και τα σχήματα 6.7, 6.8, 6.9 και 6.10. Σύμφωνα με τις εικόνες αυτές έχουμε :

- PERSONS → 0,65
- COMPANY → 0,55
- LIBRARIES → 0,5
- ISWC → 0,8

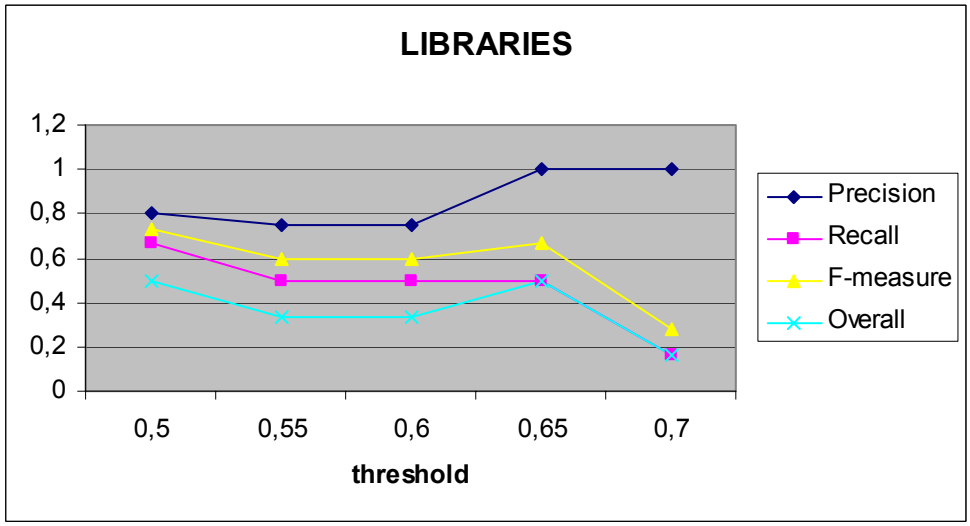
Η διαφορά που υπάρχει στο threshold που χρησιμοποιήθηκε στο σύνολο δεδομένων ISWC από τα thresholds των υπολοίπων συνόλων, οφείλεται στο γεγονός ότι τα datatype properties της προς αντιστοίχιση οντολογίας του συνόλου δεδομένων ISWC έχουν υψηλό βαθμό ομοιότητας με τα αντίστοιχα πεδία της βάσης δεδομένων του ίδιου συνόλου κάτι που δεν ισχύει στα υπόλοιπα σύνολα δεδομένων.



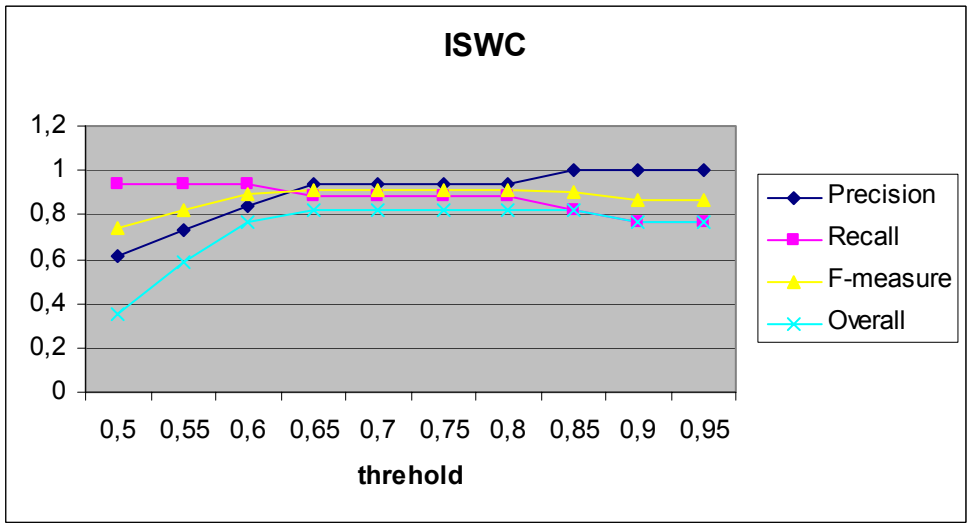
Σχήμα 6.7 Αποτελέσματα της αντιστοίχισης πεδίων σε datatype-properties της οντολογίας για το σύνολο δεδομένων “PERSONS”



Σχήμα 6.8 Αποτελέσματα της αντιστοίχισης πεδίων σε datatype-properties της οντολογίας για το σύνολο δεδομένων “COMPANY”



Σχήμα 6.9 Αποτελέσματα της αντιστοίχισης πεδίων σε datatype-properties της οντολογίας για το σύνολο δεδομένων “LIBRARIES”



Σχήμα 6.10 Αποτελέσματα της αντιστοίχισης πεδίων σε datatype-properties της οντολογίας για το σύνολο δεδομένων “ISWC”

Παρατηρούμε ότι υπάρχει μια διαφορά στα thresholds που χρησιμοποιούν οι αλγόριθμοι της αντιστοίχισης πινάκων σε κλάσεις και της αντιστοίχισης πεδίων σε datatype properties. Για παράδειγμα στο σύνολο δεδομένων LIBRARIES το threshold που χρησιμοποιείται στην αντιστοίχιση κλάσεων είναι 0,9 ενώ το αντίστοιχο για την αντιστοίχιση πεδίων σε datatype properties είναι 0,5. Η απόκλιση αυτή οφείλεται στο γεγονός ότι στα πεδία των πινάκων και στα datatype properties των οντολογιών συνήθως δε χρησιμοποιούνται έγκυρα ονόματα αλλά συνδυασμοί ονομάτων. Για παράδειγμα, είναι πολύ συχνό φαινόμενο στα datatype-properties (αλλά και στα object-properties) μιας οντολογίας να χρησιμοποιείται το πρόθεμα *has* ή το πρόθεμα *is* (π.χ. hasSalary, isMotherOf, hasWritten κτλ) για να κάνει το νόημα των properties πιο σαφές στους ανθρώπους. Συνδυασμοί λέξεων χρησιμοποιούνται και στα πεδία των πινάκων μιας βάσης, όπως για παράδειγμα student_id, birth_date, κτλ. Η χρήση λοιπόν μικρών τιμών σε μερικά thresholds δε σημαίνει και μικρή ακρίβεια στα αποτελέσματα αλλά αρκετά μεγάλη διαφορά στα ονόματα των αντίστοιχων στοιχείων.

6.5 Αξιολόγηση της απλής αντιστοίχισης κλάσεων και της αντιστοίχισης των πεδίων σε datatype properties.

Στην ενότητα αυτή συγκρίνονται τα αποτελέσματα που έδωσαν οι αλγόριθμοι της απλής αντιστοίχισης κλάσεων και της αντιστοίχισης των πεδίων σε datatype properties με τα αποτελέσματα του αλγορίθμου της εικόνας 6.2 για συγκεκριμένες τιμές στα thresholds της απλής αντιστοίχισης κλάσεων και της αντιστοίχισης των πεδίων της βάσης δεδομένων σε datatype properties της οντολογίας.

Σύμφωνα με τον αλγόριθμο της εικόνας 6.2 η αντιστοίχιση των πεδίων σε datatype properties της οντολογίας επηρεάζεται από και επηρεάζει την απλή αντιστοίχιση κλάσεων. Αυτό σημαίνει ότι αν δύο πίνακες της βάσης δεδομένων T1 και T2 αποτελούν υποψήφιες κλάσεις για την κλάση C της οντολογίας, ο αλγόριθμος θα αποφανθεί ποιος από τους T1 και T2 είναι πιθανότερο να αντιστοιχηθεί στην κλάση C μετρώντας το πλήθος των αντιστοιχίσεων των πεδίων του T1 αλλά και του T2 στα datatype properties της οντολογίας με domain την κλάση C. Όπως και στον αλγόριθμο της εικόνας 5.1, έτσι και στον εναλλακτικό αλγόριθμο της εικόνας 6.2 (ο οποίος περιέχει τους αλγορίθμους των εικόνων 5.1 και 5.18) η αντιστοίχιση των πεδίων των πινάκων της βάσης δεδομένων σε datatype properties επηρεάζεται από την αντιστοίχιση των πινάκων σε

κλάσεις με τη διαφορά ότι στον αλγόριθμο της εικόνας 6.2 τα αποτελέσματα της αντιστοίχισης των πεδίων σε datatype properties επηρεάζουν και την αντιστοίχιση των πινάκων σε κλάσεις της οντολογίας.

```
Concepts = {all concepts of the ontology}
Tables = {all tables of the database} \ {tables which represent n:m relationships}
AllDatatypePropertyMappings ← ∅
AllConceptMappings ← ∅
int max
For each concept c in Concepts do
  max = 0
  String candidateConcept
  For each table t in Tables do
    if (t is "similar" to c)  /* that is, if (t > threshold_Concept Mapping) */
    then CCS_c ← CCS_c ∪ {t}
  End for
  For each candidate concept cc in CCS_c do
    Columns_cc = {all columns of candidate concept cc} \ {foreign keys and
                 auto-increment fields}
    DP_c = {all datatype-properties with domain c or a super-concept of c}

    DatatypePropertyMappings_cc→c ← ∅
    For each column a in Columns_cc do
      For each datatype-property dp in DP_c do
        Compute the name_similarity between a and dp
        Compute the compatibility between datatype of a and the the range of dp
        If (datatype of a is equal to the the range of dp)
          then similarity = name_similarity
        else similarity = w * name_similarity + (1 – w) * compatibility
        If (similarity > threshold_Datatype Property Mapping)
         then  CDPS_dp = CDPS_dp ∪ {a}
               DatatypePropertyMappings_cc→c ← DatatypePropertyMappings_cc→c
                                               ∪{a→dp}
    End for
```

End for

If (|DatatypePropertyMappings$_{cc \rightarrow c}$ | > max)

then max = |DatatypePropertyMappings$_{cc \rightarrow c}$ |

candidateConcept ← *cc*

End for

AllDatatypePropertyMapppings ← AllDatatypePropertyMapppings ∪

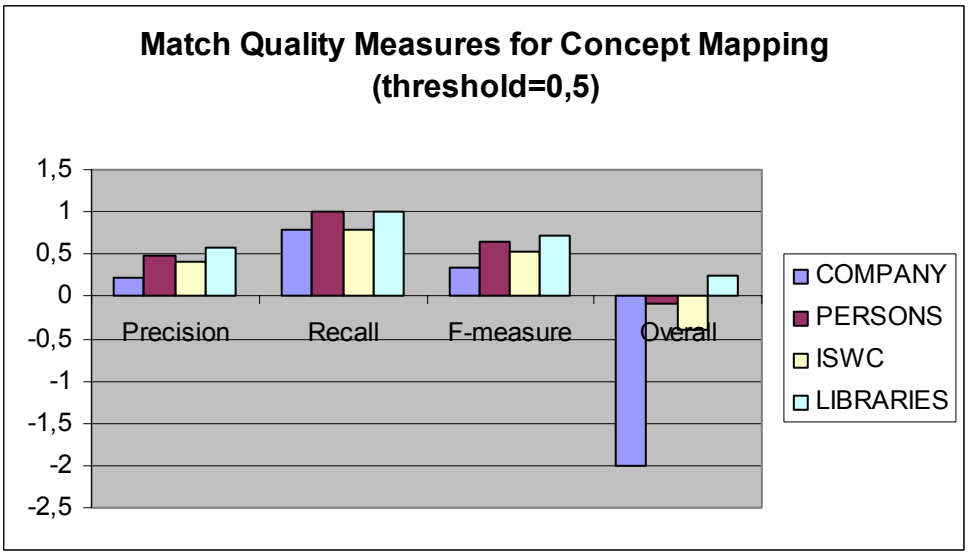
DatatypePropertyMappings$_{candidateConcept \rightarrow c}$

AllConceptMappings ← AllConceptMappings ∪ {candidateConcept→*c*}

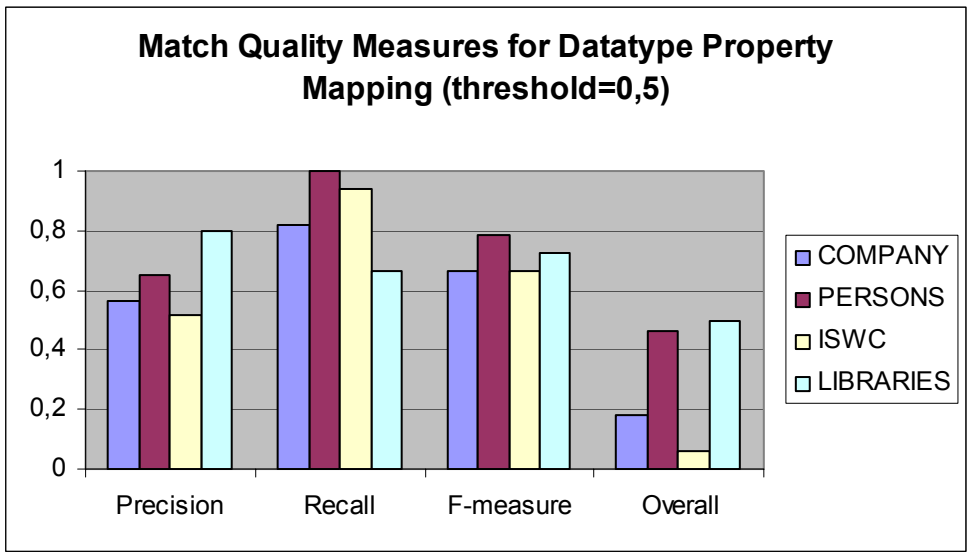
End for

Εικόνα 6.2 Εναλλακτικός αλγόριθμος για concept mapping και datatype property mapping

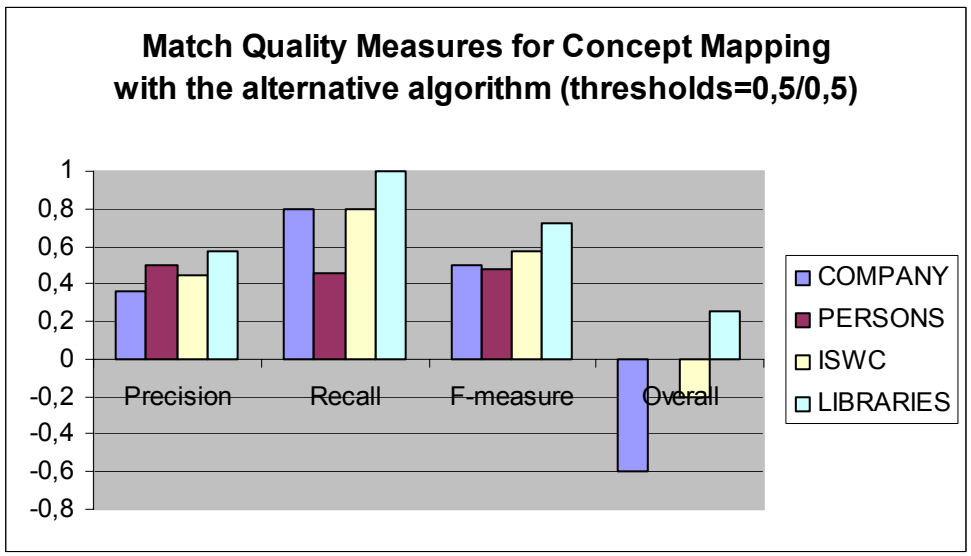
Οι αλγόριθμοι των εικόνων 5.1, 5.18 και 6.2 εφαρμόστηκαν στα τέσσερα σύνολα δεδομένων του παραρτήματος Α για συγκεκριμένες τιμές στα thresholds της απλής αντιστοίχισης κλάσεων και της αντιστοίχισης των πεδίων σε datatype properties και έδωσαν τα αποτελέσματα των σχημάτων 6.11 έως 6.16.



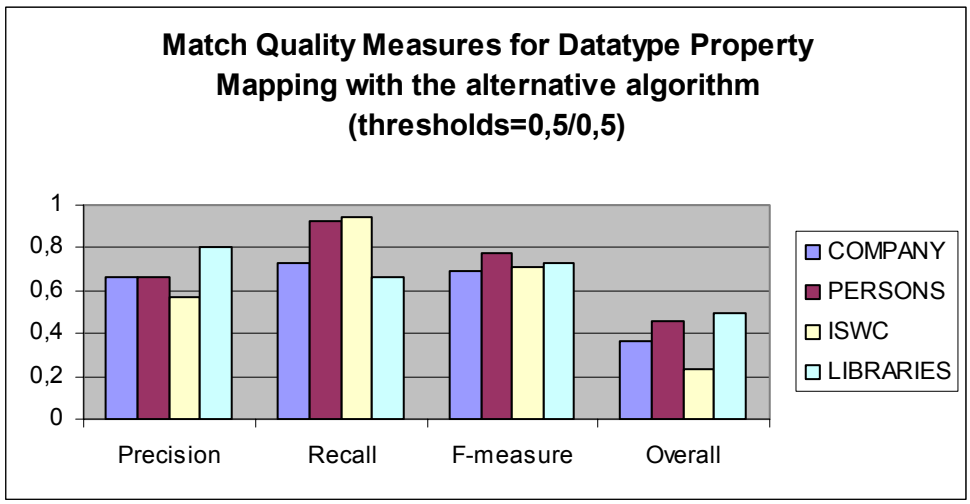
Σχήμα 6.11 Αποτελέσματα απλής αντιστοίχισης κλάσεων για threshold = 0,5.



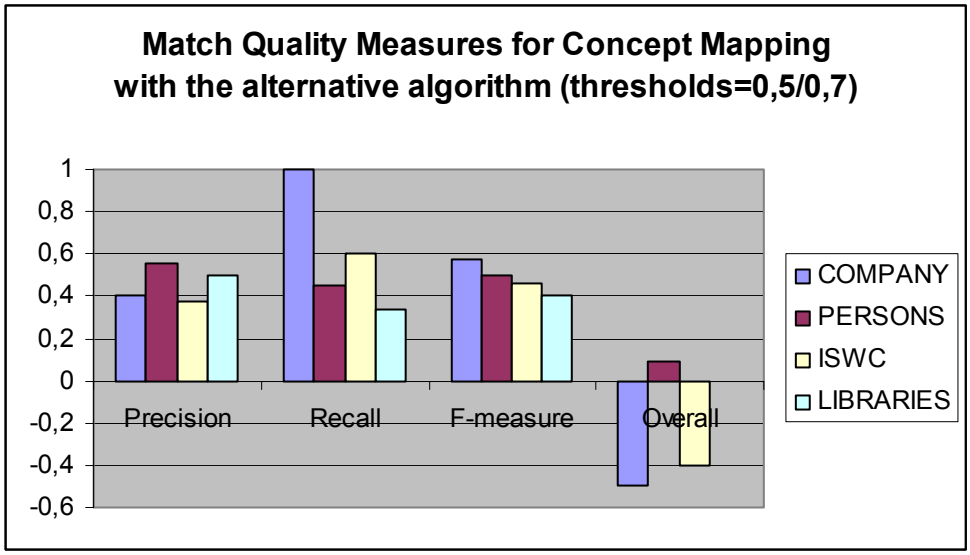
Σχήμα 6.12 Αποτελέσματα απλής αντιστοίχισης πεδίων σε datatype properties για threshold = 0,5.



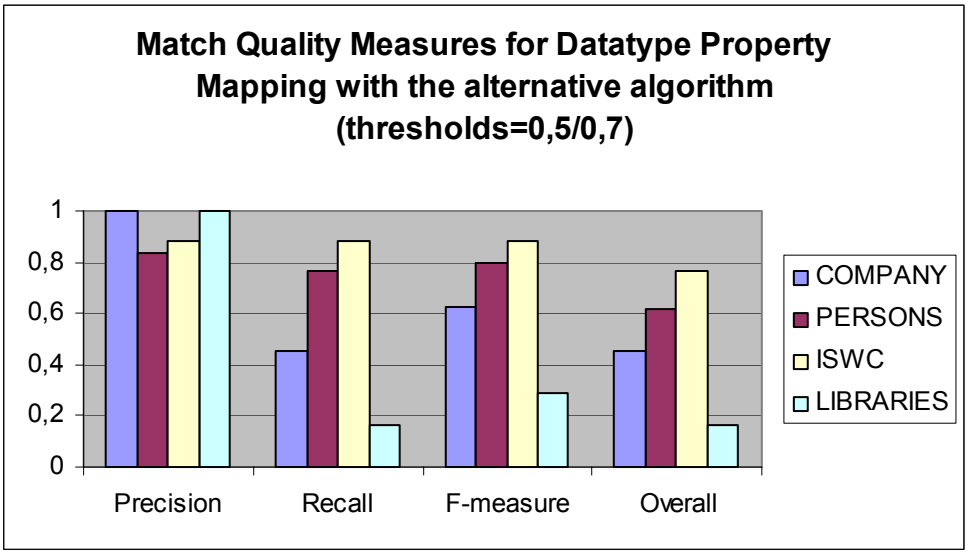
Σχήμα 6.13 Αποτελέσματα απλής αντιστοίχισης κλάσεων με τον αλγόριθμο της εικόνας 6.12 για $\text{threshold}_{\text{Concept Mapping}} = 0,5$ και $\text{threshold}_{\text{Datatype property Mapping}} = 0,5$.



Σχήμα 6.14 Αποτελέσματα αντιστοίχισης των πεδίων σε datatype properties με τον αλγόριθμο της εικόνας 6.2 για $threshold_{Concept\ Mapping}$ = 0,5 και $threshold_{Datatype\ property\ Mapping}$ = 0,5.



Σχήμα 6.15 Αποτελέσματα απλής αντιστοίχισης κλάσεων με τον αλγόριθμο της εικόνας 6.2 για $threshold_{Concept\ Mapping}$ = 0,5 και $threshold_{Datatype\ property\ Mapping}$ = 0,7.



Σχήμα 6.16 Αποτελέσματα αντιστοίχισης των πεδίων σε datatype properties με τον αλγόριθμο της εικόνας 6.2 για $\text{threshold}_{\text{Concept Mapping}} = 0,5$ και $\text{threshold}_{\text{Datatype property Mapping}} = 0,7$.

Όπως ήταν αναμενόμενο, η μείωση των thresholds στο 0,5 για τους αλγορίθμους της απλής αντιστοίχισης κλάσεων και της αντιστοίχισης των πεδίων σε datatype properties και στα τέσσερα σύνολα δεδομένων, έδωσε μεγαλύτερο πλήθος αντιστοιχίσεων με αποτέλεσμα να μειωθεί το Precision και να αυξηθεί το Recall. Στο παράρτημα Δ φαίνεται αναλυτικά πως μεταβάλλεται το πλήθος των αντιστοιχίσεων που προτείνουν οι αλγόριθμοι των εικόνων 5.1 και 5.18 με τη μεταβολή του αντίστοιχου threshold.

Στα σχήματα 6.11, 6.13 και 6.15 παρατηρούμε ότι το μέγεθος Overall παίρνει αρνητικές τιμές για κάποια σύνολα δεδομένων. Αυτό συμβαίνει γιατί το Precision είναι πολύ χαμηλό (κάτω του 0,5). Γενικά, το Overall, σε αντίθεση με τις άλλες μετρικές, μπορεί να πάρει αρνητικές τιμές αν το πλήθος των false positives είναι μεγαλύτερο από το πλήθος των true positives [37] (δηλαδή, αν $|C| > |B|$ ή αν Precision < 0,5).

Ενώ στα σχήματα 6.11 και 6.13 δεν υπάρχει σημαντική μεταβολή στα αποτελέσματα των αλγορίθμων 5.1 και 6.2, όσον αφορά την απλή αντιστοίχιση κλάσεων, παρατηρούμε ότι υπάρχει μεγάλη διαφορά στην τιμή του Recall για το σύνολο δεδομένων PERSONS. Συγκεκριμένα, η τιμή του Recall στο σύνολο δεδομένων PERSONS στο οποίο έχει εφαρμοστεί ο αλγόριθμος 6.2 είναι κατά 50% μικρότερη από την τιμή που έχει η ίδια μετρική αν εφαρμοστεί ο αλγόριθμος 5.1 στο ίδιο σύνολο δεδομένων. Η διαφορά αυτή οφείλεται στο γεγονός ότι ο αλγόριθμος 6.2, προκειμένου να αποφασίσει ποιος πίνακας της βάσης μπορεί να αντιστοιχηθεί σε μια κλάση της οντολογίας, δίνει μεγαλύτερη βαρύτητα στο πλήθος των αντιστοιχίσεων των πεδίων του

πίνακα σε datatype properties της οντολογίας, με domain την συγκεκριμένη κλάση, παρά στην ομοιότητα του ονόματος του πίνακα με το όνομα της κλάσης. Ας θεωρήσουμε για παράδειγμα την κλάση EmailAddress της οντολογίας του συνόλου δεδομένων PERSONS. Από τους πίνακες addresses και emailaddresses της βάσης δεδομένων του ίδιου συνόλου που αποτελούν υποψήφιες κλάσεις για την κλάση EmailAddress, ο αλγόριθμος 6.2 επιλέγει τον πίνακα addresses καθώς τα πεδία του ταιριάζουν περισσότερο με τα datatype properties της οντολογίας με domain την κλάση EmailAddress από ότι ταιριάζουν τα πεδία του πίνακα emailaddresses. Το γεγονός αυτό οδηγεί στο να προτείνει ο αλγόριθμος μια εσφαλμένη αντιστοίχιση και να παραλείψει την πραγματική.

Στα σχήματα 6.15 και 6.16 φαίνεται ο τρόπος με τον οποίο επηρεάζεται η απλή αντιστοίχιση κλάσεων και η αντιστοίχιση των πεδίων των πινάκων σε datatype properties της οντολογίας αντίστοιχα, όταν αυξηθεί το threshold της αντιστοίχισης των πεδίων σε datatype properties από την τιμή 0,5 στην τιμή 0,7. Όσον αφορά την απλή αντιστοίχιση κλάσεων, η αύξηση αυτή του threshold δεν προκαλεί σημαντικές μεταβολές στο Precision, γεγονός που δεν ισχύει για το Recall (σχήματα 6.13 και 6.15). Στο σύνολο δεδομένων LIBRARIES παρατηρούμε τη μεγαλύτερη μείωση του Recall. Η αύξηση του threshold οδήγησε σε μεγάλη μείωση των πραγματικών αντιστοιχίσεων των πεδίων σε datatype properties για το συγκεκριμένο σύνολο, με αποτέλεσμα ο αλγόριθμος 6.2 να δώσει και μικρότερο πλήθος πραγματικών αντιστοιχίσεων πινάκων σε κλάσεις της οντολογίας. Στο σύνολο δεδομένων COMPANY, η αύξηση του threshold μείωσε σημαντικά το πλήθος των λανθασμένων αντιστοιχίσεων και όχι τόσο το πλήθος των πραγματικών αντιστοιχίσεων των πεδίων σε datatype properties, με αποτέλεσμα να αυξηθεί, όπως δείχνει το σχήμα 6.15, ο αριθμός των πραγματικών αντιστοιχίσεων των πινάκων σε κλάσεις της οντολογίας. Τέλος, στα υπόλοιπα δύο σύνολα δεδομένων η αύξηση του threshold δεν προκάλεσε σημαντικές μεταβολές στο datatype property mapping με αποτέλεσμα να μην επηρεαστεί σημαντικά και το concept mapping.

ΚΕΦΑΛΑΙΟ 7

ΕΠΙΛΟΓΟΣ

Ο Σημασιολογικός Ιστός είναι η νέα υποδομή του Παγκοσμίου Ιστού που θα επιτρέψει την επεξεργασία του περιεχομένου του ιστού από τους υπολογιστές και θα αποτελέσει τη λύση πολλών μειονεκτημάτων του σημερινού ιστού. Είναι λοιπόν αναγκαία η δημιουργία κοινά-διαμοιραζόμενης σημασιολογικής πληροφορίας. Στην παρούσα εργασία παρουσιάστηκε ένας τρόπος ταιριάσματος δύο διαφορετικών μοντέλων δεδομένων.

Συγκεκριμένα, παρουσιάστηκε μια μεθοδολογία η οποία βρίσκει ημι-αυτόματα αντιστοιχίσεις μεταξύ των στοιχείων του σχήματος μιας βάσης δεδομένων και των στοιχείων μιας οντολογίας Σημασιολογικού Ιστού με απώτερο στόχο τον εμπλουτισμό της οντολογίας με σημασιολογικά δεδομένα προερχόμενα από τα σχεσιακά. Η διαδικασία εύρεσης αντιστοιχίσεων ανάμεσα στα στοιχεία δύο σχημάτων είναι μια εξαιρετικά ενδιαφέρουσα διαδικασία αν σκεφτεί κανείς τις διαφορές που υπάρχουν στους περιορισμούς των προς αντιστοίχιση σχημάτων και το γεγονός ότι τα σχήματα μπορεί να έχουν κατασκευαστεί από διαφορετικούς χρήστες.

Ωστόσο, υπάρχουν ανοικτά θέματα που αφορούν στην αντιστοίχιση των στοιχείων δύο σχημάτων η επίλυση των οποίων θα κάνει τη διαδικασία της αντιστοίχισης πιο αποτελεσματική.

Μέχρι τώρα τα πεδία των πινάκων μιας βάσης δεδομένων αντιμετωπίστηκαν ως ατομικά. Θα ήταν χρήσιμο να μπορέσουμε να «διασπάσουμε» σύνθετα γνωρίσματα στα επιμέρους συστατικά. Αυτό απαιτείται, για παράδειγμα, στην περίπτωση που το πεδίο *address* ενός πίνακα μιας βάσης δεδομένων με τιμές της μορφής «Μυκόνου 11 Ανθούσα, 15349» πρέπει να αντιστοιχηθεί στα datatype properties *street*, *city* και *zipcode*.

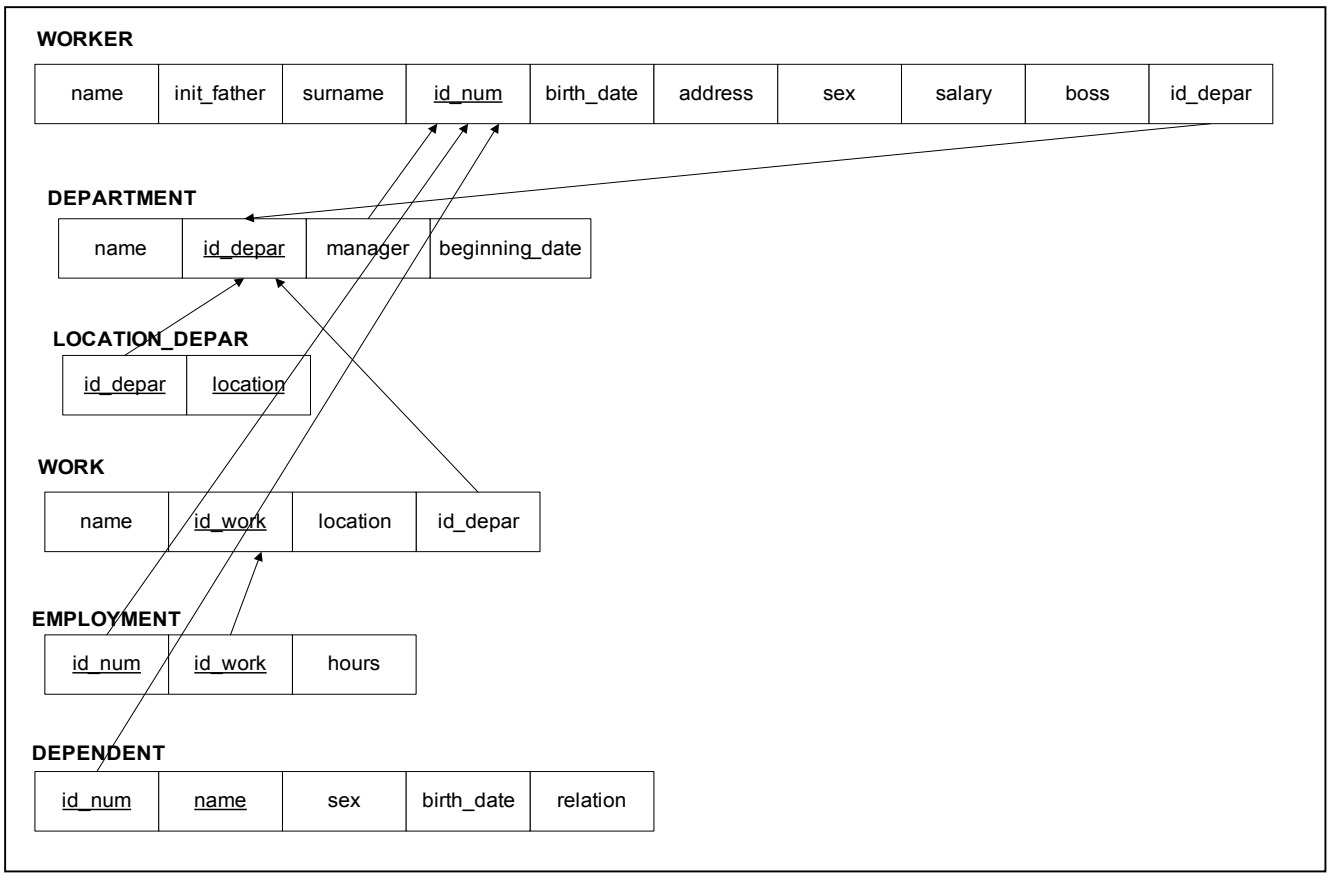
Στην μεθοδολογία που αναπτύχθηκε στην παρούσα εργασία προτάθηκε αλγόριθμος αντιστοίχισης των πινάκων της βάσης δεδομένων που αναπαριστούν N:M σχέσεις, ανάμεσα σε δύο άλλους πίνακες, σε object properties της οντολογίας. Σε μια βάση δεδομένων όμως ενδέχεται να υπάρχουν πίνακες που να αναπαριστούν v-αδικές (v>2) σχέσεις ανάμεσα σε άλλους πίνακες κάτι που δεν ισχύει στις γλώσσες που χρησιμοποιούνται στον Σημασιολογικό Ιστό (RDF(S), OWL) αφού ένα property (είτε datatype ή object) αποτελεί μια δυαδική σχέση. Η εύρεση λοιπόν του τρόπου αντιστοίχισης πινάκων της βάσης δεδομένων που αναπαριστούν v-αδικές σχέσεις σε properties της οντολογίας θα έκανε τη διαδικασία της αντιστοίχισης αποτελεσματικότερη.

Τέλος, η ένδειξη περιγραφών (explanations) στα στοιχεία των σχημάτων με τη βοήθεια λεξικού αλλά και η αυτόματη αλλαγή των παραμέτρων της αντιστοίχισης (π.χ. thresholds) θα συντελούσε στην αντιμετώπιση των ασαφειών και στην κατανόηση των αποτελεσμάτων αντίστοιχα.

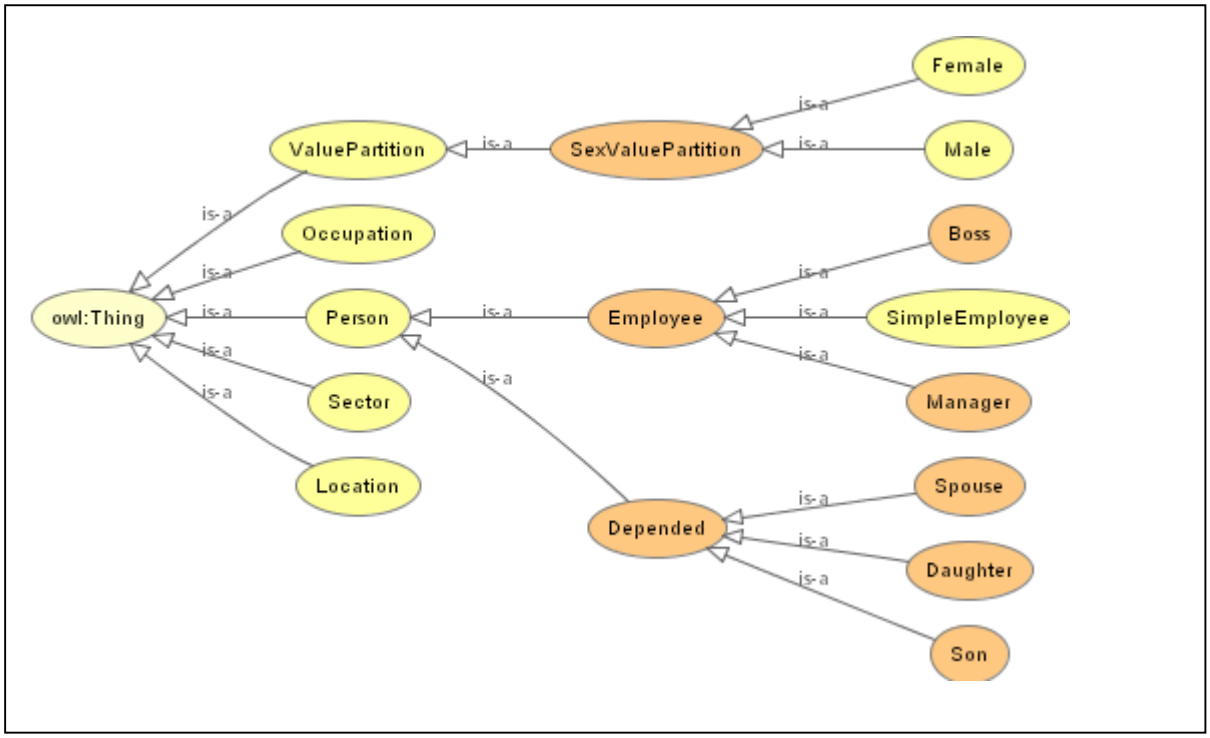
ΠΑΡΑΡΤΗΜΑ Α

Στο Παράρτημα αυτό δίνονται τα σχήματα των βάσεων και οι ιεραρχίες κλάσεων των οντολογιών που χρησιμοποιήθηκαν στην τεκμηρίωση των αλγορίθμων. Αναλυτικός ορισμός των σχεσιακών σχημάτων και των οντολογιών στις γλώσσες SQL και OWL αντίστοιχα υπάρχει μέσα στο CD-ROM.

Σύνολο δεδομένων "COMPANY"

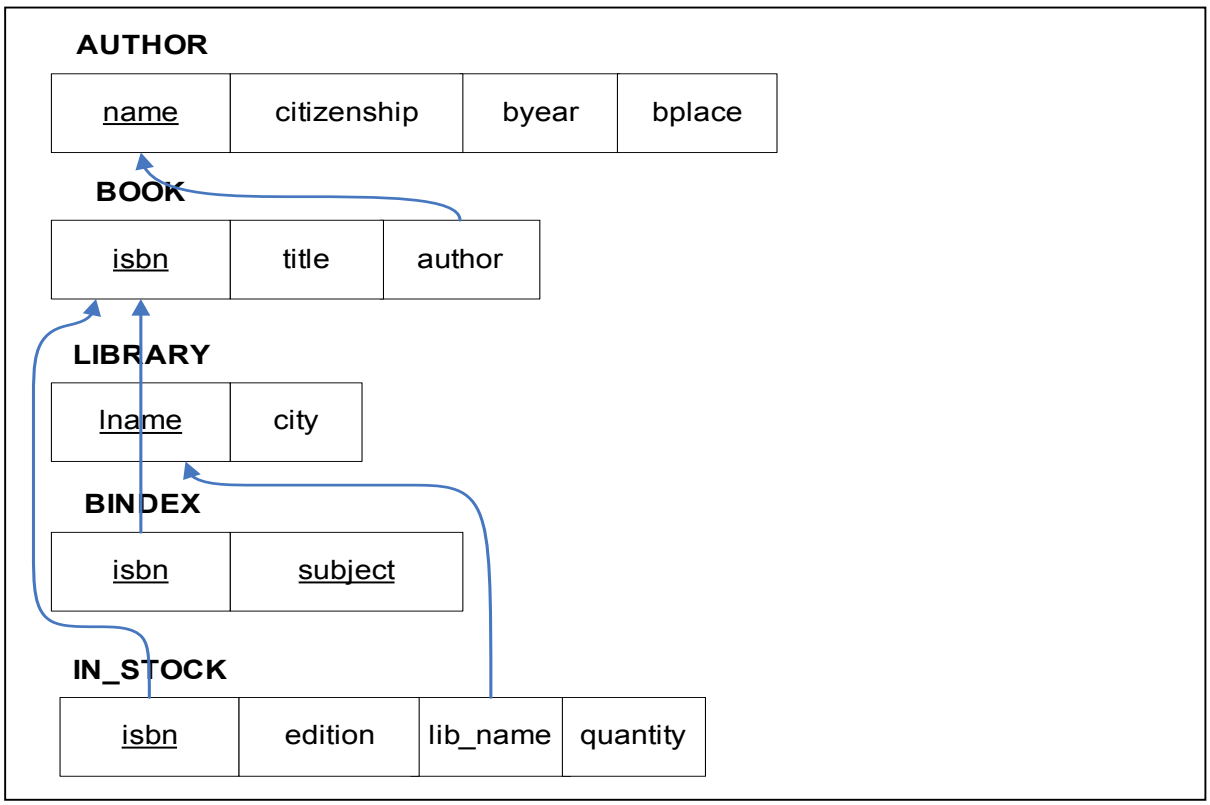


Σχήμα Βάσης δεδομένων "COMPANY"

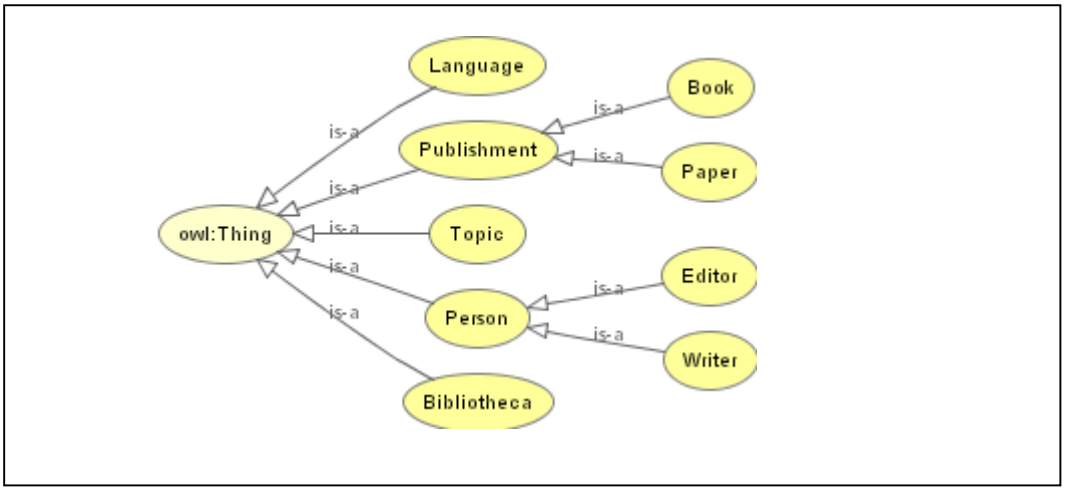


Ιεραρχία κλάσεων της οντολογίας "COMPANY"

Σύνολο δεδομένων "LIBRARIES"

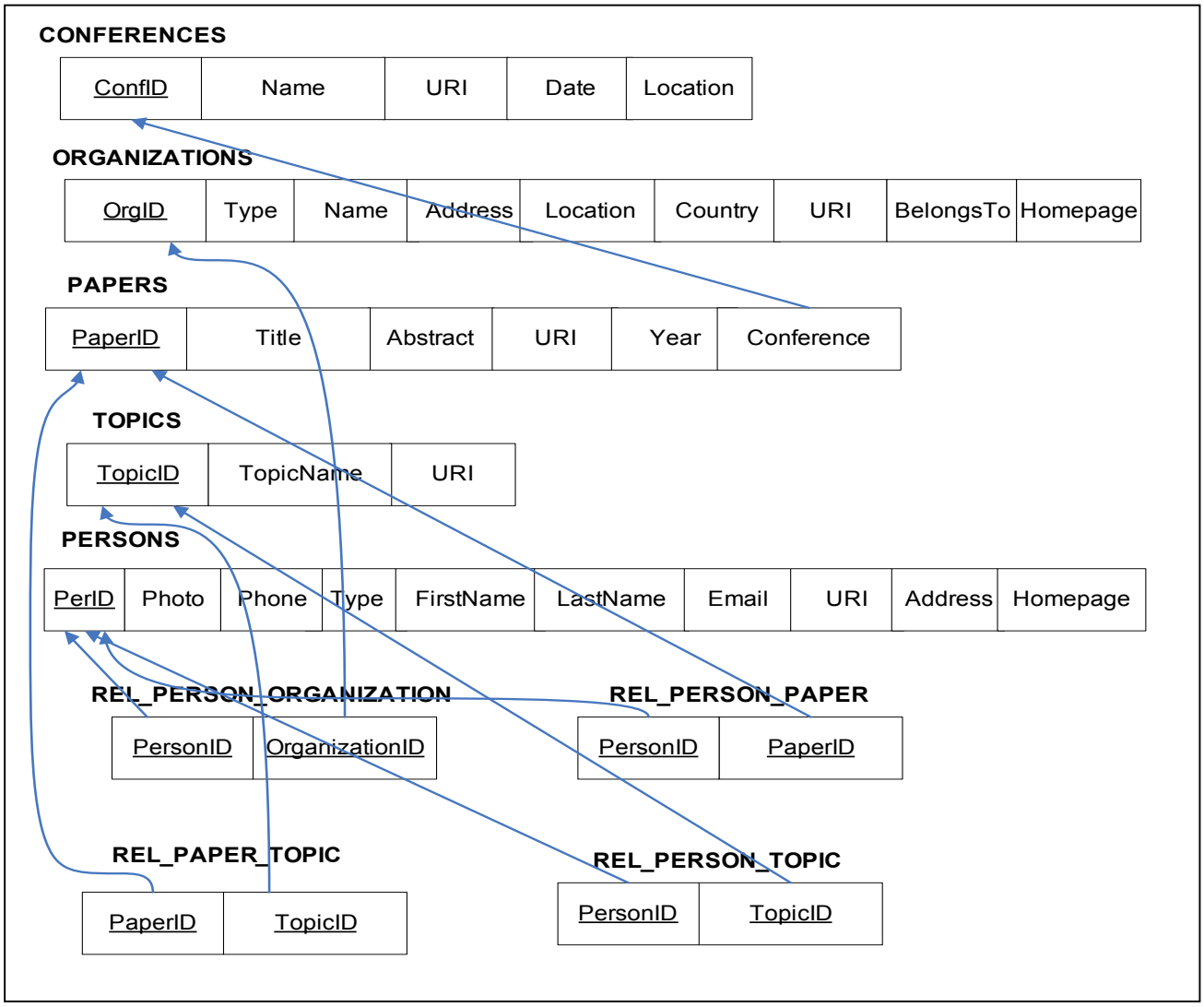


Σχήμα Βάσης δεδομένων "LIBRARIES"

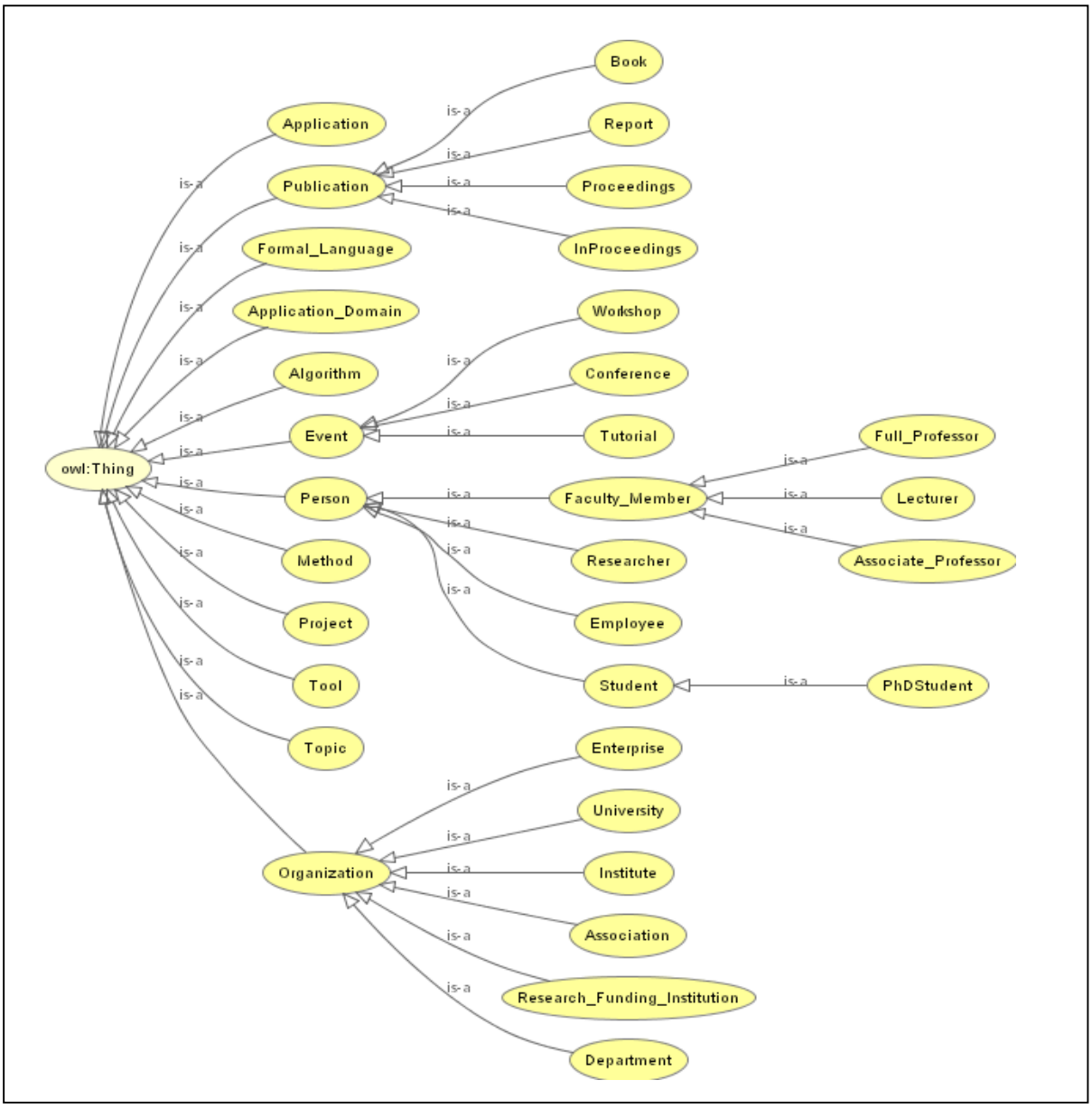


Ιεραρχία κλάσεων της οντολογίας "LIBRARIES"

Σύνολο δεδομένων "ISWC"

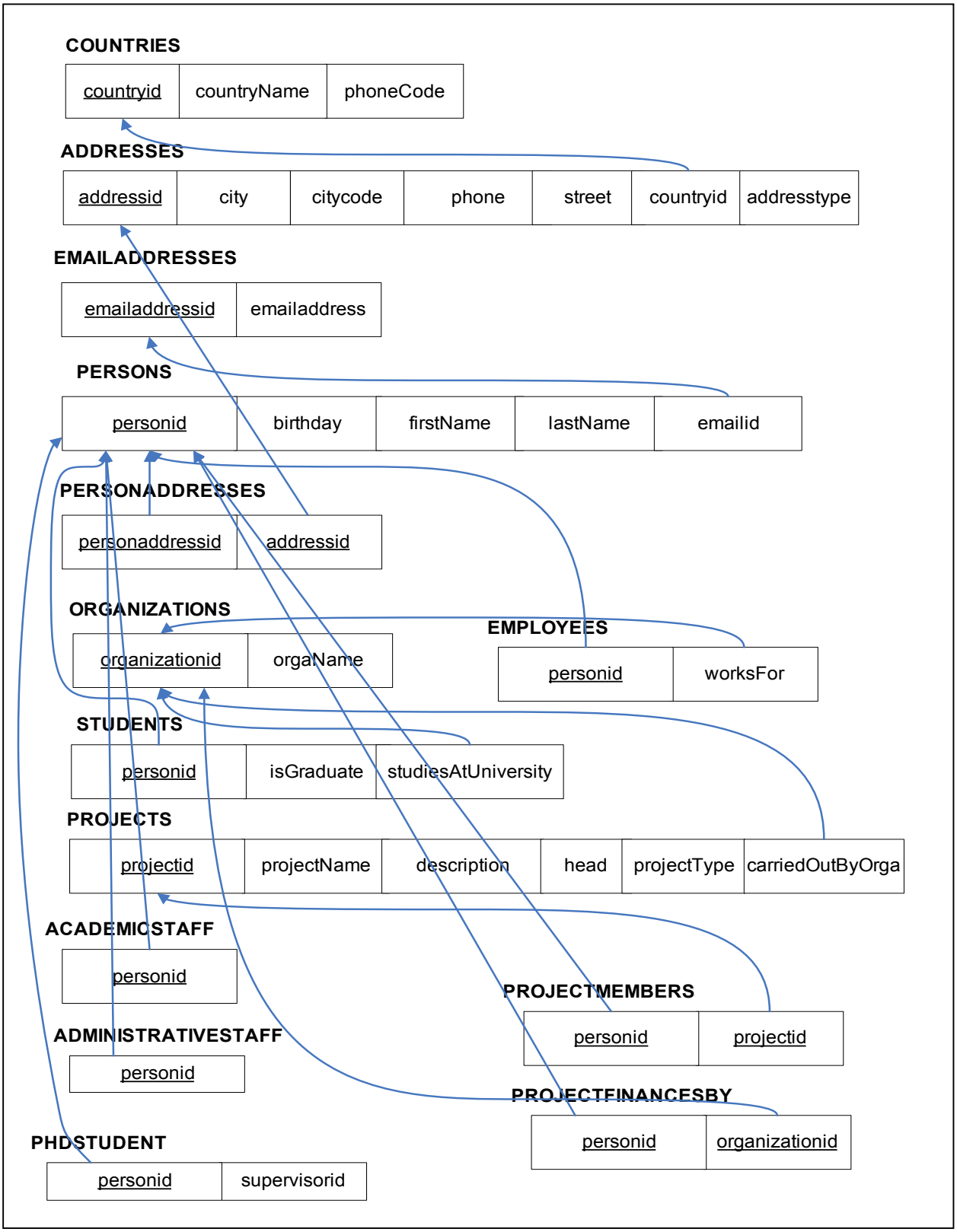


Σχήμα Βάσης δεδομένων "ISWC"

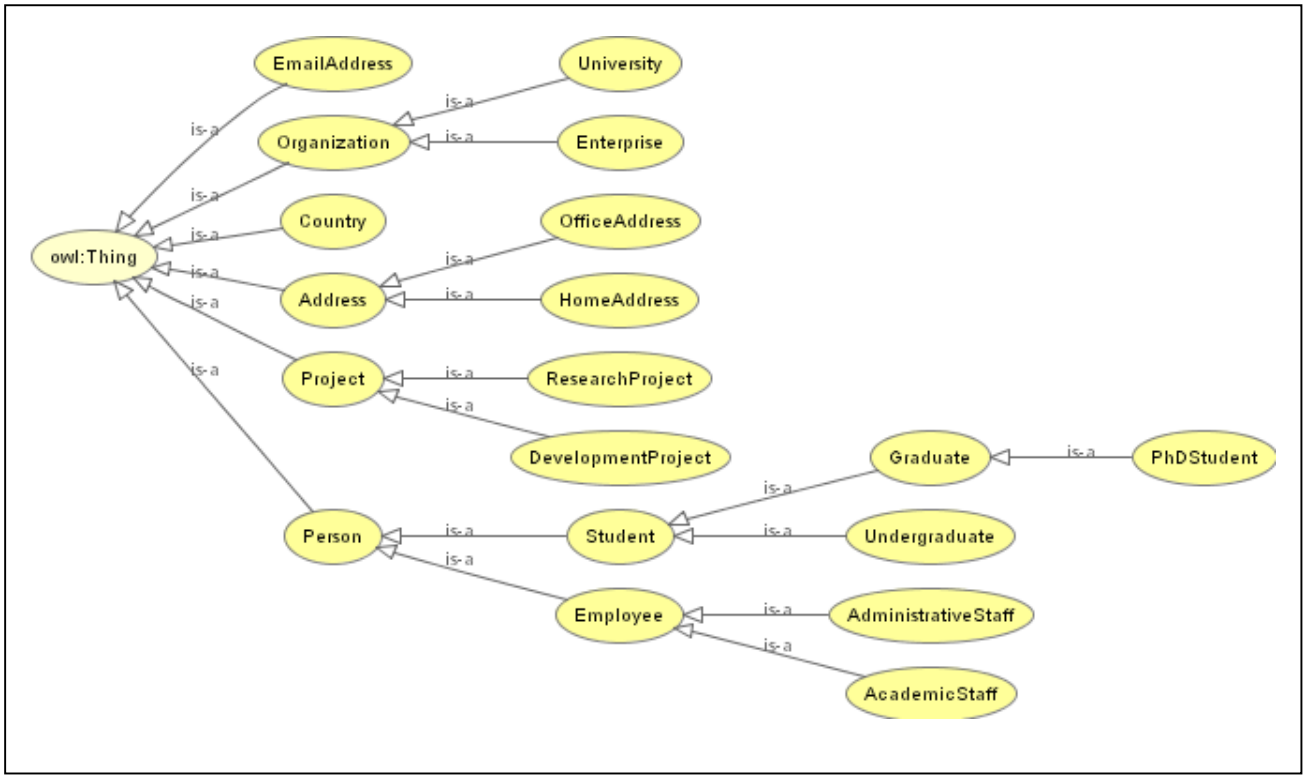


Ιεραρχία κλάσεων της οντολογίας "ISWC"

Σύνολο δεδομένων "PERSONS"



Σχήμα Βάσης δεδομένων "PERSONS"



Ιεραρχία κλάσεων της οντολογίας "PERSONS"

ΠΑΡΑΡΤΗΜΑ Β

Στο παράρτημα αυτό δίνονται οι πραγματικές αντιστοιχίσεις που ισχύουν ανάμεσα στα σχήματα του παραρτήματος Α.

Σύνολο δεδομένων "COMPANY"

Table to Concept Mappings	
Tables	Concepts
worker	Worker
department	Department
work	Work
dependent	Depended

Columns to Datatype Properties Mappings	
Columns	Datatype Properties
worker.birth_date	hasBirthDate
worker.salary	hasSalary
department.beginning_date	hasStartDate
Worker.address	livesAt
worker.name	firstName
worker.surname	lastName
worker.init_father	fatherInitial
department.name	name
Work.name	name
dependent.name	firstName
dependent.birth_date	hasBirthDate

Σύνολο δεδομένων "PERSONS"

Table to Concept Mappings	
Tables	Concepts
countries	Country
addresses	Address
emailaddresses	EmailAddress
persons	Person
organizations	Organization
students	Student
employees	Employee
academicstaff	AcademicStaff
phdstudent	PhDStudent
projects	Project
administrativeStaff	AdministrativeStaff

Columns to Datatype Properties Mappings	
Columns	Datatype Properties
organizations.orgaName	name
projects.projectName	name
countries.countryName	name
addresses.phone	phone
countries.phoneCode	phoneCode
addresses.street	street
addresses.lastName	lastName
addresses.firstName	firstName
projects.description	description
addresses.city	city
addresses.cityCode	cityCode
persons.birthday	birthday
emailaddresses.emailAdress	address

Σύνολο δεδομένων "ISWC"

Table to Concept Mappings	
Tables	Concepts
conferences	Conference
organizations	Organization
persons	Person
topics	Topic
papers	InProceedings

Columns to Datatype Properties Mappings	
Columns	Datatype Properties
organizations.Address	Address
persons.address	address
organizations.Country	country
conferences.Date	date
persons.Email	email
conference.name	eventTitle
persons.FirstName	first_Name
persons.Homepage	homepage
persons.LastName	last_Name
conference.Location,	location
organization.Location	location
organization.name	name
topics.topicName	name
persons.Phone	phone
persons.Photo	photo
papers.Title	title
papers.Year	year

Σύνολο δεδομένων "LIBRARIES"

Table to Concept Mappings	
Tables	Concepts
author	Writer
book	Book
library	Bibliotheca

Columns to Datatype Properties Mappings	
Columns	Datatype Properties
author.byear	hasBirthYear
author.name	hasName
book.title	hasTitle
author.citizenship	livesIn
library.lname	hasName
library.city	isLocatedIn

ΠΑΡΑΡΤΗΜΑ Γ

Στο Παράρτημα αυτό δίνονται τα αποτελέσματα των αλγορίθμων για κάθε ένα από τα σύνολα δεδομένων.

Σύνολο δεδομένων “COMPANY”

Threshold (concept mapping) = 0,85

Threshold (datatype-property mapping) = 0,55

Table	Concept	Degree of Similarity
department	Sector	0,92307692
dependent	Depended	0,88888888
work	Occupation	0,85714285
worker	Employee	0,92307692
worker	Person	0,90909090

Column	Datatype-property	Degree of Similarity
department.name	Name	1,0
dependent.name	lastName	0,64375
dependent.name	firstName	0,60606
dependent.birth_date	hasBirthDate	0,70416
work.name	Name	1,0
worker.salary	hasSalary	0,77350
worker.name	lastName	0,64375
worker.name	firstName	0,60606
worker.init_father	fatherInitial	0,63286
worker.surname	lastName	0,57451
worker.birth_date	hasBirthDate	0,70416

Σύνολο δεδομένων “ISWC”

Threshold (concept mapping) = 0,8

Threshold (datatype-property mapping) = 0,8

Table	Concept	Degree of Similarity
conferences	Conference	0,90550239
organizations	Organization	0,92056856
papers	Report	0,93333333
persons	Person	0,84740259
topics	Topic	0,81944444

Column	Datatype-property	Degree of Similarity
conferences.date	date	1,0
conferences.location	location	1,0
organizations.name	name	1,0
organizations.address	address	1,0
organizations.location	location	1,0
organizations.country	country	1,0
papers.title	title	1,0
papers.abstract	title	0,857142
papers.year	year	1,0
persons.firstname	first_Name	0,836764
persons.lastname	last_Name	0,816666
persons.address	address	1,0
persons.email	email	1,0
persons.homepage	homepage	1,0
persons.phone	phone	1,0
persons.photo	photo	1,0

Σύνολο δεδομένων “PERSONS”

Threshold (concept mapping) = 0,8

Threshold (datatype-property mapping) = 0,65

Table	Concept	Degree of Similarity
academicstaff	AcademicStaff	0,96153846
Addresses	Address	0,86111111
administrativestaff	Administrativestaff	0,97368421
emailaddress	EmailAddress	0,89285714
emloyees	Emloyee	0,88333333
organisations	Organizations	0,81438127
persons	Person	0,84740259
phdstudent	PhdStudent	0,925
projects	Project	0,86778846
students	Student	0,86778846

Column	Datatype-property	Degree of Similarity
addresses.addressid	address	0,654168
addresses.city	city	1,0
addresses.citycode	cityCode	0,975000
addresses.phone	phone	1,0
addresses.street	street	1,0
emailaddresses.emailaddress	address	0,748774
organizations.organame	name	0,675000
persons.birthday	birthday	1,0
persons.firstname	lastName	0,722222
persons.firstname	firstName	0,972222
persons.lastname	lastName	0,968500
persons.lastname	firstName	0,722222
projects.description	description	1,0

Σύνολο δεδομένων “LIBRARIES”

Threshold (concept mapping) = 0,9

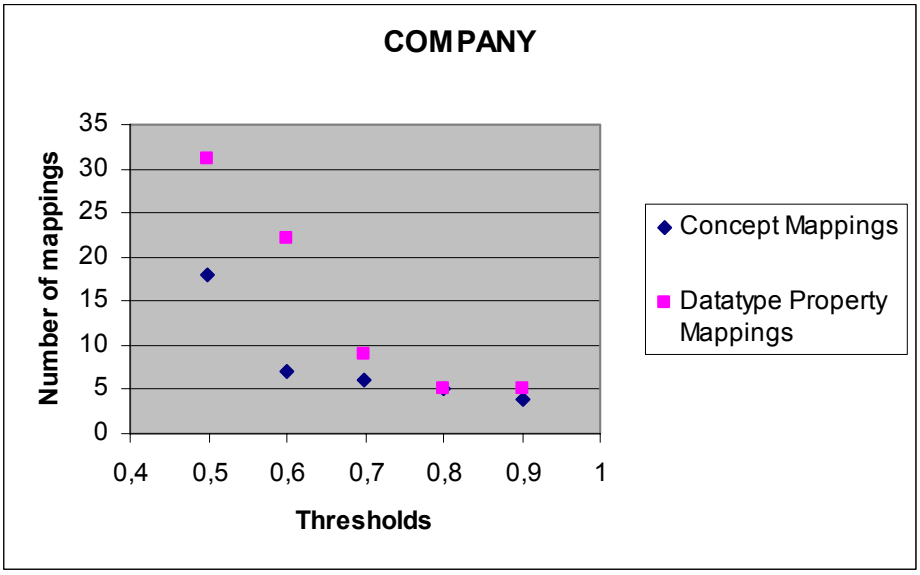
Threshold (datatype-property mapping) = 0,5

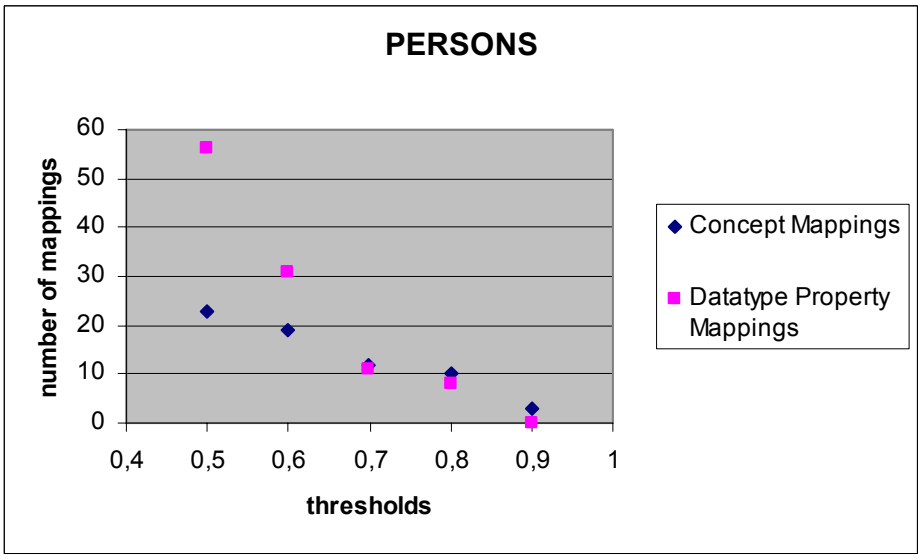
Table	Concept	Degree of Similarity
author	Writer	1,0
book	Book	1,0
library	Bibliotheca	0,93333333

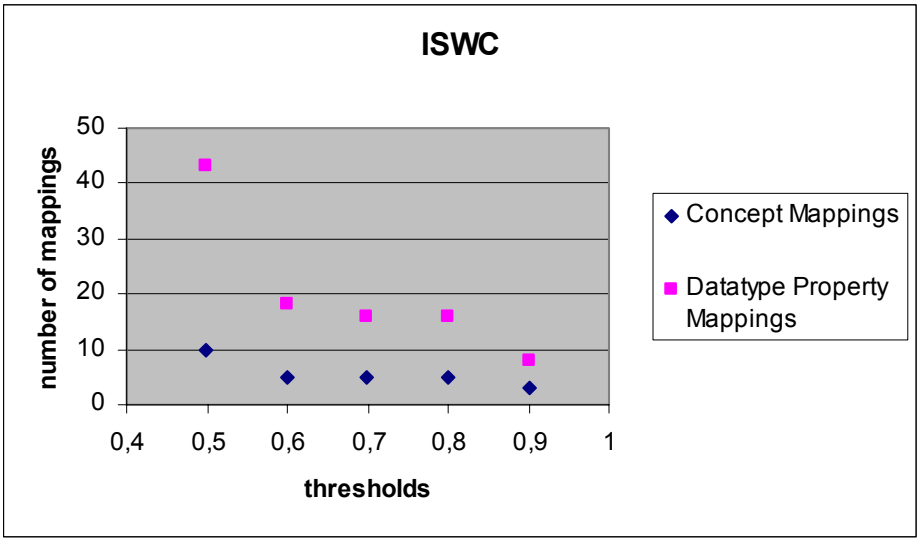
Column	Datatype-property	Degree of Similarity
author.name	hasName	0,690476
author.byear	hasBirthYear	0,512500
book.title	edition	0,629579
book.title	hasTitle	0,738636
library.lname	hasName	0,657142

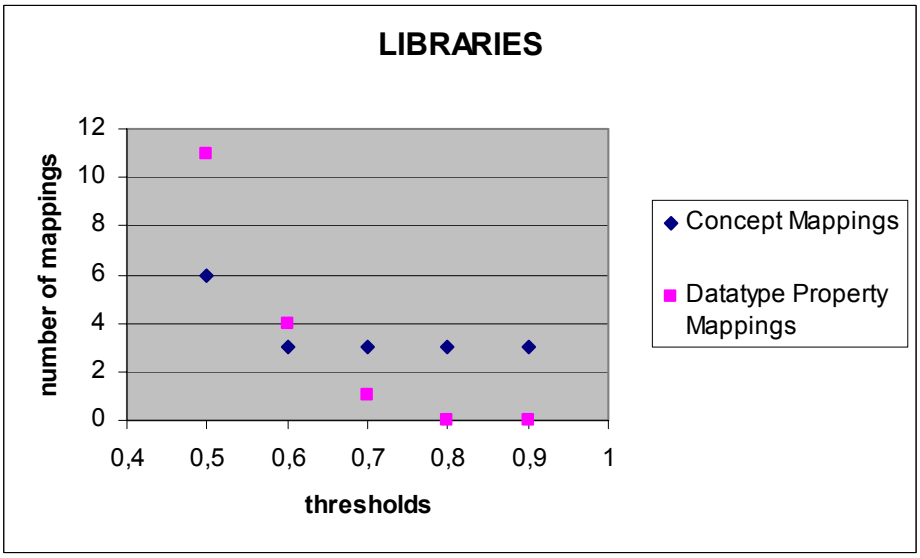
ΠΑΡΑΡΤΗΜΑ Δ

Στο παράρτημα αυτό δίνονται οι γραφικές παραστάσεις μεταβολής του πλήθους των αντιστοίχισεων που προτείνουν οι αλγόριθμοι της απλής αντιστοίχισης κλάσεων και της αντιστοίχισης των πεδίων σε datatype properties σε σχέση με τη μεταβολή του αντίστοιχου threshold για κάθε ένα από τέσσερα σύνολα δεδομένων του παραρτήματος Α.






ISWC
number of mappings
50
40
30
20
10
0
0,4
0,5
0,6
0,7
0,8
0,9
1
thresholds
Concept Mappings
Datatype Property Mappings


LIBRARIES
number of mappings
12
10
8
6
4
2
0
0,4
0,5
0,6
0,7
0,8
0,9
1
thresholds
Concept Mappings
Datatype Property Mappings

ΑΚΡΩΝΥΜΙΑ

JDBC	Java DataBase Connector
SQL	Structured Query Language
RDF	Resource Description Framework
OWL	Web Ontology Language
RDBMS	Relational Database Management Systems
XML	Extensible Markup Language
RDQL	RDF Data Query Language
ER	Entity Relationship
D2R	Database to Relational
ER	Entity Relationship
R2O	Relational to Ontology
IC	Integrity Constraints
WWW	World Wide Web
GUI	Graphical User Interface
HTML	HyperText Markup Language
W3C	World Wide Web Consortium
SW	Semantic Web
CCS	Candidate Concept Set
CC	Candidate Concept
CDP	Candidate Datatype Property
CDPS	Candidate Datatype Property Set
COP	Candidate Object Property
COPS	Candidate Object Property Set

ΑΝΑΦΟΡΕΣ

1. R. Elmasri, S. B. Navathe, “Fundamentals of Database Systems”, Second Edition, Benjamin Cummings, 1994
2. Barak Naveh and Contributors, A class library that provides mathematical graph-theory objects and algorithms, http://jgrapht.sourceforge.net. July 2005.
3. XML Schema - Datatypes Quick Reference, http://www.xml.dvint.com, 2002, 2003, D Vint Productions
4. Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborn, and Kevin Wilkinson, “Jena: Inplementing the Semantic Web Recommendations”, Technical Report, HP Labs, 2003.
5. Dave Raggett, Arnaud Le Hors, Ian Jacobs, “HyperText Markup Language (HTML) 4.01”, W3C Recommendation, 24 December 1999.
6. Nenad Stojanovic, Ljiljana Stojanovic, Raphael Volz , “A reverse engineering approach for migrating data-intensive web sites to the Semantic Web”, In proceedings of the Conference on Intelligent Intformation Processing, World Computer Congress, Montreal, Canada, 2002, Kluwer, Academic, Publishers
7. Raphael Volz, Siegfried Handschuh, Steffen Staab, Ljiljana Stojanovic, Nenad Stojanovic, “Unveiling the hidden bribe: deep annotation for mapping and migrating legacy data to the Semantic Web”, Web Semantics: Science, Services and Agents on the World Wide Web 1 (2004) 187-206.
8. Christian Vizer, “D2Rmap – A Database to RDF Mapping Language”, Poster at the 12th World Wide Web Conference, Budapest, Hungary, 2003.
9. Tim Bray, Eve Maler, Jean Paoli, and C. M. Sperberg-McQueen, “Extensible Markup Language (XML) 1.0”, W3C Recommendation, 6 October 2000.
10. Jayant Madhavan, Philip A. Bernstein, “Generic Schema Matching with Cupid”, in Proc. Of the 27th VLDB. 49-58, 2001.
11. Christian Bizer, and Andy Seabone, “D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs. 3rd International Semantic Web Conference, Hirosima, Japan, 2004

12. RDQL: A Query Language for RDF, http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/

13. Joseki, A SPARQL Server for Jena, http://www.joseki.org/

14. Erhard Rahm, Philip A. Bernstein, "A survey of approaches to automatic schema mapping", The VLDB Journal 10:334-350(2001)

15. Asunción Gómez-Pérez, Mariano Fernández-López, Oscar Corcho, "Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web", (Advanced Information and Knowledge Processing), Springer, 2004

16. TR. Gruber, "A Translation Approach to Portable Ontology Specification", Knowledge Acquisition, 5(2):199-220, 1993.

17. N. Guarino, "Formal Ontology in Information Systems", In Guarino (ed) First International Conference on Formal Ontology in Information Systems (FOIS '98), Trento, Italy, IOS Press, Amsterdam, pp 3-15, 1998.

18. Swartout B, Ramesh P, knight K, Russ T (1997) "Toward Distributed Use of large-Scale Ontologies". In: Farquhar A, Gruninger M, Gómez-Pérez A, Uschold M, van der Vet P (eds) AAAI'97 Spring Symposium on Ontological Engineering. Stanford University, California, pp 138-148

19. The World Wide Web Consortium, http://www.w3.org/

20. Do, H.H., E.Rahm, "COMA – A System for Flexible Combination of Schema Matching Approach", VLDB 2002

21. Protégé, http://protege.stanford.edu/

22. Φώτω Αφράτη, Γιώργος Παπαγεωργίου, «Αλγόριθμοι: Μέθοδοι Σχεδίασης και Ανάλυση Πολυπλοκότητας», Εκδόσεις Συμμετρία, Αθήνα 1993

23. J. Barrasa, O. Corcho, A. Gomez-Perez, "R2O, an Extensible and Semantically Based Database-to-Ontology Mapping Language", Second Workshop on Semantic Web and Databases (SWDB2004). Toronto, Canada. August 2004.

24. Melnik S., H. Garcia-Molina, E.Rahm, "Similarity Flooding: A Versatile Graph Matching Algorithm", ICDE 2002

25. V. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals", Soviet Physics – Doklady 10, 10:707—710, 1966.

26. M. A. Jaro, “Advances in record linking methodology as applied to the 1985 census of Tampa Florida”, Journal of the American Statistical Society, vol. 64, pp 1183-1210, 1989.

27. S. B. Needleman, C. D. Wunch, “Needleman-Wunch Algorithm for Sequence Similarity Searches”, Journal of Mol. Biology, vol. 48, pp 443-453, 1970.

28. L. Gravano, P. G. Ipeirotis, H. V. Jagadish, N. Koudas, S. Muthukrishnan, L. Pietarinen, D. Srivastava, “Using q-grams in a DBMS for Approximate String Processing”, IEEE Data Engineering Bulletin, vol. 24, No 4, pp 28-34, 2001.

29. D. Lin, “An Information-Theoretic Definition of Similarity”, In Proceedings of the 15th International Conf. on Machine Learning, pp 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

30. Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, “XML Schema Part 1: Structures”, W3C Recommendation, 2 May 2001.

31. G. Antoniou, F. van Harmelen, “A Semantic Web Primer”, MIT Press, Massachusetts, 2004.

32. A Semantic Network database for English, Princeton University, http://wordnet.princeton.edu/

33. Wikipedia: the free encyclopedia, http://en.wikipedia.org/

34. C. Leacock, M. Chodorow, “Combining local context and WordNet similarity for word sense identification”, In C. Fellbaum, editor, WordNet: An Electronic Lexical Database, pp 265–283, MIT Press, 1998.

35. Z. Wu, M. Palmer, “Verb semantics and lexical selection”, In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp 133–138, Las Cruces, New Mexico, 1994.

36. Tim Berners-Lee, James Hendler, Ora Lassila, “The Semantic Web”, Scientific American, May 2001

37. Hong-Hai Do, Sergey Melnik, Erhand Rahm, “Comparison of Schema Matching Evaluations” Proc. GI-Workshop “Web and Databases”, Erfurt, Oct. 2002

38. Jeremy J. Carroll, and Graham Kline, “Resource Description Framework (RDF): Concepts and Abstract Syntax”, W3C Recommendation, 10 February 2004.

39. Tim Berners-Lee, Robert Cailliau, Jean-Francois Groff, Bernd Pollermann, "World-Wide Web: The Information Universe", CERN.

40. Google, http://www.google.com

41. Yahoo, http://www.yahoo.com

42. Altavista, http://www.altavista.com

43. Guarino N, Giaretta P (1995) "Ontologies and Knowledge Bases: Towards a Terminological Clarification" In Mars N (ed) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (KBKS' 95). University of Twente, Enschede, The Netherlands. IOS Press, Amsterdam, the Netherlands.

44. Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, Chris Wroe, "A Practical Guide to Building OWL Ontologies Using the Protégé-OWL Plugin and CO-ODE Tools" Edition 1.0, The University of Manchester, August 27, 2004

45. Frank Van Harmelen, and Deborah L. McGuinness, "OWL Web Ontology Language Overview", W3C Recommendation, 10 February 2004.

46. WebODE, A Scalable Workbench for Ontological Engineering, http://webode.dia.fi.um.es/webode/

47. OntoEdit: An Ontology Engineering Environment supporting the development and maintenance of ontologies using graphical means http://www.ontoknowledge.org/tools/ontoedit.shtml

48. Rogers Cadenhead, Laura Lemay, "Teach Yourself JAVA 2 in 21 Days", SAMS, third edition.

49. Michael K. Bergman, "The Deep Web: Surfacing hidden value." White paper, July 2000.

50. B. S. Lee, G. Wiederhold, "Outer Joins and Filters for Instantiating Objects form Relational Databases through Views", IEEE Trans. On Knowledge Engineering (TKDE), Vol. 6, No. 1, February 1994.

51. Κωνσταντίνος Κ. Κολομβάτσος, "Συγκριτική Αξιολόγηση Μεθόδων Λεξικογραφικής και Σημασιολογικής Ομοιότητας", Δπλωματική Εργασία, Τμήμα Πληροφορικής και Τηλεπικοινωνιών ΕΚΠΑ, Σεπτέμβριος 2005.